# Cài đặt searchBar. Màu nền…

func setSearchBar() {

if let textfield = searchBar.value(forKey: "searchField") as? UITextField {

textfield.backgroundColor = UIColor.clear

textfield.attributedPlaceholder = NSAttributedString(string: textfield.placeholder ?? "", attributes: [NSAttributedString.Key.foregroundColor : UIColor.white])

if let leftView = textfield.leftView as? UIImageView {

leftView.image = leftView.image?.withRenderingMode(.alwaysTemplate)

leftView.tintColor = UIColor.white

}

}

}

# Loại bỏ icon search trong searchBar

searchBar.setImage(UIImage(), for: .search, state: .normal)

You probably also want to adjust the gap on the left side:

searchBar.setPositionAdjustment(UIOffset(horizontal: -20, vertical: 0), for: .search)

# lưu ý khi làm việc khi làm việc với google map

hàm chạy vào khi dừng scroll bản đồ

func mapView(\_ mapView: GMSMapView, idleAt position: GMSCameraPosition) {

// Từ màn tra cứu chung sang thì không call findByMapFrame ngay lần đầu. Chỉ call khi đã scroll map

if isChangeCable { // đang tinh chỉnh cáp

if indexChangeCable != nil {

if let indexChangeCable = indexChangeCable {

let temp = listMarkerCable[indexChangeCable].userData as? String

if temp != "noChange", isVatCap == false {

// không call hàm findNearest khi vừa tap marker chỉnh cáp

if firstTapChangeTap {

firstTapChangeTap = false

} else {

apiType = .findNearest

setValueParams()

findNearest(params: params)

}

}

}

markerCableOfCableLane()

}

}

guard pauseCallAPI == false else { return }

if callByMapFrameDefault {

callAPIWhenStopScrollMap()

}

switch fromVC {

case .OtherVC: // từ màn chi tiết của các màn sang thì gọi lần đầu

// chạy lần 1 để đổi giá trị. Không gọi api khi vừa vào màn này, gọi khi vuốt bản đồ.

callByMapFrameDefault = true

case .TraCuuHaTangNimsVC:

callByMapFrameDefault = true

default:

callAPIWhenStopScrollMap()

}

}

Hàm chạy vào khi scroll map

func mapView(\_ mapView: GMSMapView, didChange position: GMSCameraPosition) {

guard position.zoom > 14 else {

return

}

// Cancel all request

if apiType == .findCableLaneByMapFrame {

}

Alamofire.SessionManager.default.session.getTasksWithCompletionHandler { (sessionDataTask, uploadData, downloadData) in

print("Cancel all request")

sessionDataTask.forEach { $0.cancel() }

uploadData.forEach { $0.cancel() }

downloadData.forEach { $0.cancel() }

self.hideLoading()

}

let projection = mapView.projection.visibleRegion()

let topLeftCorner: CLLocationCoordinate2D = projection.farLeft

let bottomRightCorner: CLLocationCoordinate2D = projection.nearRight

x1 = topLeftCorner.latitude

y1 = topLeftCorner.longitude

x2 = bottomRightCorner.latitude

y2 = bottomRightCorner.longitude

// latitude = mapView.camera.target.latitude

// longtitude = mapView.camera.target.longitude

guard isChangeCable else { return }

guard let currentCableChange = currentCableChange else { return }

guard let count = currentCableChange.geometry?.coordinates.count else { return }

guard indexChangeCable != 0, indexChangeCable != count - 1 else { return }

centerMapCoordinate = position.target

if let indexChangeCable = indexChangeCable {

currentCableChange.geometry?.coordinates[indexChangeCable].latitude = centerMapCoordinate.latitude

currentCableChange.geometry?.coordinates[indexChangeCable].longitude = centerMapCoordinate.longitude

}

if let indexPolyline = indexPolyline {

listCable[indexPolyline] = currentCableChange

for item in listPolyLine {

if let i = listPolyLine.index(of: item) {

listPolyLine[i].map = nil

listPolyLine.remove(at: i)

totalPolyline.remove(at: i)

}

}

isVatCap = false

drawCableRoute()

markerCableOfCableLane()

}

}

Cả 2 hàm sẽ cùng chạy vào khi load map lên view, trong đó hàm didChange position sẽ chạy vào trước hàm idleAt position