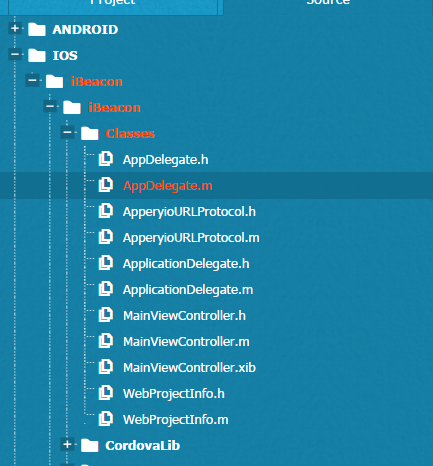
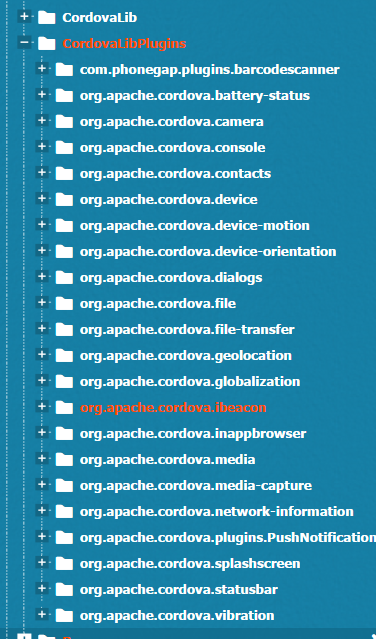
**Adding IBeacon Cardova Plugin:**

Replace AppDelegate.m file

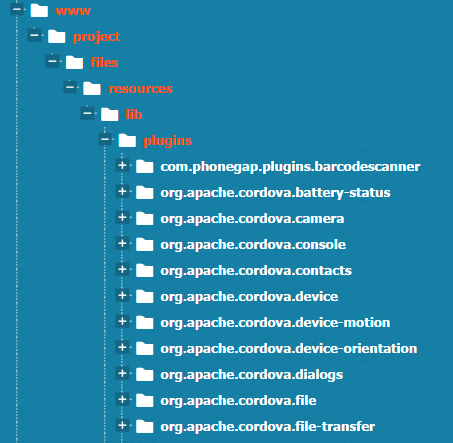
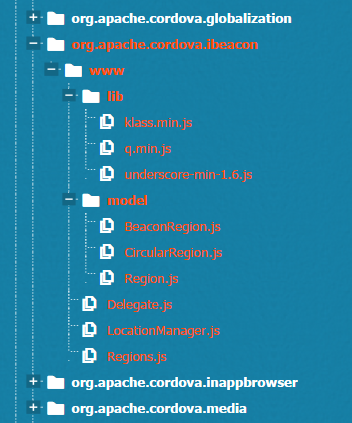


Create package in CordovaLibPlugins with name org.apache.cordova.ibeacon Add files to package :

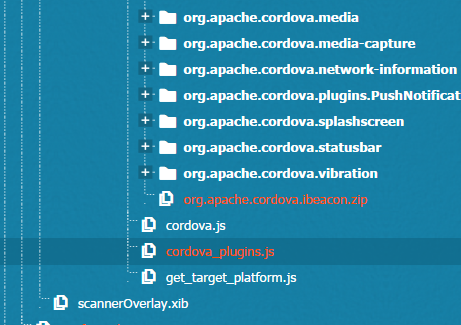
 

Add Cordova plug-in java script files under Resources :

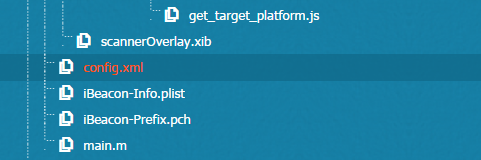
Create org.apache.cordova.ibeacon package add plugin files to package:

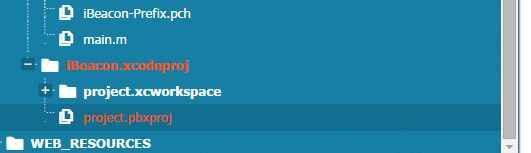
Replace cordova\_plugins.js with file content:



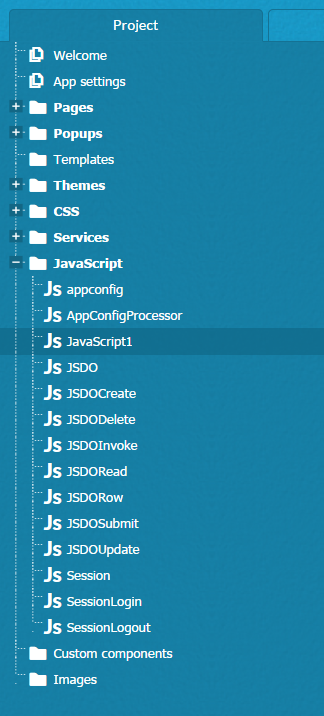
Add custom cordova plug-in details in config.xml:



Replace project.pbxproj file :



Add java script file to project to call IBeacon :



Add script in Java Script:

var on = new Object();

**create IBeacon object for searching beacon with beacon Id:**

function createBeacon(uuid, identifier) {

var minor = 3333; // optional, defaults to wildcard if left empty

var major = 1000; // optional, defaults to wildcard if left empty

var beaconRegion = new locationManager.locationManager.BeaconRegion(identifier, uuid, major, minor);

return beaconRegion;

}

**Initialize location manager listener for searching beacon with beacon Id:**

function initial() {

var delegate = new locationManager.locationManager.Delegate().implement({

didDetermineStateForRegion: function(pluginResult) {

locationManager.locationManager.appendToDeviceLog('[DOM] didDetermineStateForRegion: ' + JSON.stringify(pluginResult));

},

didStartMonitoringForRegion: function(pluginResult) {

}

didRangeBeaconsInRegion: function(pluginResult) {

rangeresponse = JSON.parse(JSON.stringify(pluginResult));

var uuid = rangeresponse["region"]["uuid"].toUpperCase();

var identifier = rangeresponse["region"]["identifier"];

var distance = rangeresponse["beacons"][0]["proximity"];

var rssi = rangeresponse["beacons"][0]["rssi"];

**Make ajax call for data when Ibeaocn Detected in IBeacon near Range:**

if(distance.toUpperCase() == String("proximityNear").toUpperCase()) {

if (on[uuid] == 0) {

on[uuid] = 1;

$.ajax({

url: "https://www.rollbase.com/rest/api/selectQuery?query=select beaconmessage from beacon where beaconuid='"+uuid+"'&loginName=ram1023&password=Progress123&custId=106603520",

type: 'GET',

dataType: 'html',

success: function(data, textStatus, xhr) {

start = data.indexOf("<col>") + 5;

end = data.indexOf("</col>");

var data1 = data.substring(start,end);

alert(data1);

},

error: function(xhr, textStatus, errorThrown) {

alert('error');

}

});

}

} else if (distance.toUpperCase() == String("proximityFar").toUpperCase()) {

if (on[uuid] == 1) {

on[uuid] = 0;

//alert("far");

//document.getElementById("MRegion").innerHTML = "Far "+uuid;

//$("#MRegion").append("Far -> " + uuid + "<br>");

}

}

//locationManager.locationManager.stopRangingBeaconsInRegion(createBeacon(uuid, identifier));

}

});

locationManager.locationManager.setDelegate(delegate);

}

**Ondevice Ready call Master() .initialize Beacon ids for searching :**

function Master() {

// alert("Master");

var uuids = new Array();

var identifiers = new Array();

var beaconRegion = new Array();

//alert("beacon");

**Add beacons for searching:**

uuids.push('86C7185B-6A99-4A38-86E4-859CA3E742A7');

uuids.push('F7028248-68B4-4F65-8087-D5D5CB3A1CBD');

for (var i = 0; i < uuids.length; i++) {

on[uuids[i]] = 0;

}

initial();

identifiers.push('Qualcomm Beacon');

identifiers.push('Progress Beacon');

for (var i = 0; i < uuids.length; i++) {

beaconRegion[i] = createBeacon(uuids[i], identifiers[i]);

//alert(beaconRegion[i]);

locationManager.locationManager.startMonitoringForRegion(beaconRegion[i]);

//locationManager.locationManager.startRangingBeaconsInRegion(beaconRegion[i]);

// locationManager.locationManager.stopRangingBeaconsInRegion(beaconRegion[i]);

}

setInterval(function() {

for (var i = 0; i < uuids.length; i++) {

//alert("123");

locationManager.locationManager.startRangingBeaconsInRegion(beaconRegion[i]);

}

}, 1000);

}