

```

import UIKit
import ArcGIS

let Coast_url = AGSArcGISMapImageLayer(url:
URL(string:"https://services.gisqatar.org.qa/server/rest/
services/Vector/Coast/MapServer"!)
let Satellite_url = AGSArcGISTiledLayer(url:
URL(string:"https://services.gisqatar.org.qa/server/rest/
services/Imagery/QatarSatellite/MapServer"!)
let Hybrid_EN_url = AGSArcGISTiledLayer(url:URL(string:
"https://services.gisqatar.org.qa/server/rest/services/
Vector/Qatar_StreetMap_Hybrid_E/MapServer"!)
let Hybrid_AR_url = AGSArcGISTiledLayer(url:URL(string:
"https://services.gisqatar.org.qa/server/rest/services/
Vector/Qatar_StreetMap_Hybrid_Ar_Test/MapServer"!)
let StreetMap_EN_url =
AGSArcGISTiledLayer(url:URL(string: "https://
services.gisqatar.org.qa/server/rest/services/Vector/
Qatar_StreetMap_E/MapServer"!)
let StreetMap_AR_url =
AGSArcGISTiledLayer(url:URL(string: "https://
services.gisqatar.org.qa/server/rest/services/Vector/
Qatar_StreetMap_A/MapServer"!)

class MapViewController: UIViewController, BasemapType {

    @IBOutlet weak var mapView: AGSMapView!

    var map: AGSMap!
    var basemapName: String!

    override var prefersStatusBarHidden: Bool {
        return true
    }

    override func viewDidLoad() {
        super.viewDidLoad()

        self.map = AGSMap(basemap: AGSBasemap(baseLayer:
Coast_url))
        self.mapView.map = self.map
        self.SetMapExtent()

```

```

        basemapName = "Coast"
    }

    func SetMapExtent () {
        let envelope = AGSEnvelope (xMin:
155000.602394999, yMin: 297009.112383, xMax:
246023.663305001, yMax: 495960.811646, spatialReference:
self.mapView.spatialReference)
        self.map.initialViewpoint =
AGSViewpoint(targetExtent: envelope)
        self.mapView.map = map
    }

    // MARK: - Navigation

    override func prepare(for segue: UIStoryboardSegue,
sender: Any?) {
        // dismiss any shown view controllers
        dismiss(animated: false)

        if segue.identifier == "basemapSegue" {
            guard let bmVC = segue.destination as?
BaseMapViewController else { return }
            bmVC.delegate = self
            bmVC.basemapName = self.basemapName
        }

        if let controller = segue.destination as?
BaseMapViewController {
            //pop over settings
            controller.presentationController?.delegate =
self
            //preferred content size
            if traitCollection.horizontalSizeClass
== .regular,
                traitCollection.verticalSizeClass
== .regular {
                controller.preferredContentSize =
CGSize(width: 250, height:300)
            } else {

```

```

        controller.preferredContentSize =
CGSize(width: 250, height: 300)
    }
}

func BasemapNamePass(bmName: String) {
    self.basemapName = bmName
    print("2:", basemapName ?? "")

    if basemapName == "Satellite" {

        self.Satellite_Map_Tapped()
    }
    else if basemapName == "Hybrid" {

        self.Hybrid_Map_Tapped()
    }
    else if basemapName == "Normal" {

        self.Normal_Map_Tapped()
    }
    else if basemapName == "Coast" {

        self.Coast_Map_Tapped()
    }

}

func Satellite_Map_Tapped(){
    let currentExtent =
self.mapView.visibleArea!.extent
    self.map = AGSMap(basemap: AGSBasemap(baseLayer:
Satellite_url))
    let envelope = AGSEnvelope (xMin:
currentExtent.xMin, yMin: currentExtent.yMin, xMax:
currentExtent.xMax, yMax: currentExtent.yMax,
spatialReference: self.mapView.spatialReference)
    self.map.initialViewpoint =
AGSViewpoint(targetExtent: envelope)
    self.mapView.map = self.map
}

func Hybrid_Map_Tapped() {

```

```

        let currentExtent =
self.mapView.visibleArea!.extent
        self.map = AGSMap(basemap: AGSBasemap(baseLayer:
Hybrid_EN_url))
        let envelope = AGSEnvelope (xMin:
currentExtent.xMin, yMin: currentExtent.yMin, xMax:
currentExtent.xMax, yMax: currentExtent.yMax,
spatialReference: self.mapView.spatialReference)
        self.map.initialViewpoint =
AGSViewpoint(targetExtent: envelope)
        self.mapView.map = self.map
    }

    func Normal_Map_Tapped() {
        let currentExtent =
self.mapView.visibleArea!.extent
        self.map = AGSMap(basemap: AGSBasemap(baseLayer:
StreetMap_EN_url))
        let envelope = AGSEnvelope (xMin:
currentExtent.xMin, yMin: currentExtent.yMin, xMax:
currentExtent.xMax, yMax: currentExtent.yMax,
spatialReference: self.mapView.spatialReference)
        self.map.initialViewpoint =
AGSViewpoint(targetExtent: envelope)
        self.mapView.map = self.map
    }

    func Coast_Map_Tapped() {
        let currentExtent =
self.mapView.visibleArea!.extent
        self.map = AGSMap(basemap: AGSBasemap(baseLayer:
Coast_url))
        let envelope = AGSEnvelope (xMin:
currentExtent.xMin, yMin: currentExtent.yMin, xMax:
currentExtent.xMax, yMax: currentExtent.yMax,
spatialReference: self.mapView.spatialReference)
        self.map.initialViewpoint =
AGSViewpoint(targetExtent: envelope)
        self.mapView.map = self.map
    }
}

extension MapViewController:
UIAdaptivePresentationControllerDelegate {

```

```
    func adaptivePresentationStyle(for controller:
UIPresentationController, traitCollection:
UITraitCollection) -> UIModalPresentationStyle {
        return .none
    }
}
```