

```

import UIKit
import ArcGIS
import Charts

//Census Map service
let Census_url = URL(string: "https://khazna.gisqatar.org.qa/fed/rest/services/
GISUDBAP/QSA_Census_District_2020/MapServer/0")

class ViewController: UIViewController, UITableViewDataSource,
UITableViewDelegate {

    @IBOutlet weak var pieChart: PieChartView!

    @IBOutlet weak var menStepper: UIStepper!
    @IBOutlet weak var womenStepper: UIStepper!

    @IBOutlet weak var tableView: UITableView!
    @IBOutlet weak var spinner: UIActivityIndicatorView!

    private var Census_FeatureTable: AGSServiceFeatureTable?
    var queried_CensusFeatures = [AGSFeature]()

    var menDataEntry = PieChartDataEntry(value: 0)
    var womenDataEntry = PieChartDataEntry(value: 0)

    var numberOfDownloadDataEntries = [PieChartDataEntry]()

    override func viewDidLoad() {
        super.viewDidLoad()

        //Accessing Census Data from Map Services

        self.Census_FeatureTable = AGSServiceFeatureTable(url: Census_url!)
        AGSAuthenticationManager.shared().delegate = self

        //Chart definitions

        pieChart.chartDescription.text = ""
        menDataEntry.value = menStepper.value
        menDataEntry.label = "Men"
        womenDataEntry.value = womenStepper.value
        womenDataEntry.label = "Women"
        numberOfDownloadDataEntries = [menDataEntry, womenDataEntry]
        updateChartData()

    }

    @IBAction func QueryCensusData(_ sender: UIButton){

        self.Query_Census()

    }

    func Query_Census() {

        self.spinner.startAnimating()

        let queryParams = AGSQueryParameters()

```

```

        queryParams.whereClause = "1=1"

        self.Census_FeatureTable?.populateFromService(with: queryParams,
clearCache: true, outFields: ["*"]) { [weak self] (queryResult:
AGSFeatureQueryResult?, error: Error?) in
            guard let self = self else { return }

            if let error = error {
                self.presentAlert(error: error)
            } else if let features = queryResult?.featureEnumerator().allObjects
{
                if !features.isEmpty {
                    self.queried_CensusFeatures = features

                    let no = self.queried_CensusFeatures.count
                    print("Number: \(String(describing: no))")

                    DispatchQueue.main.async {
                        self.tableView.reloadData()
                    }
                } else {
                    self.presentAlert(message: "No features found")
                }

                self.spinner.stopAnimating()
            }
        }
    }

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int)
-> Int {
        var count = 0
        count = self.queried_CensusFeatures.count
        return count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "Cell", for:
indexPath)

        cell.textLabel?.font = UIFont.systemFont(ofSize: 12.0)
        cell.detailTextLabel?.font = UIFont.systemFont(ofSize: 12.0)
        cell.detailTextLabel?.textColor = UIColor.brown
        let feature = self.queried_CensusFeatures[indexPath.row]
        cell.textLabel?.text = feature.attributes.value(forKey: "ENAME") as?
String
        cell.detailTextLabel?.text = String(feature.attributes.value(forKey:
"DIST_NO") as! Int)

        return cell
    }

    @IBAction func changeMen(_ sender: UIStepper) {
        menDataEntry.value = sender.value
        updateChartData()
    }

```

```

@IBAction func changeWomen(_ sender: UIStepper) {
    womenDataEntry.value = sender.value
    updateChartData()
}

func updateChartData() {
    let chartDataSet = PieChartDataSet (entries:
numberOfDownloadDataEntries, label:"")
    let chartData = PieChartData(dataSet: chartDataSet)

    let colors = [UIColor(named: "menColor"), UIColor(named: "womenColor")]
    chartDataSet.colors = colors as! [NSUIColor]

    pieChart.data = chartData
}

}

extension ViewController: AGSAuthenticationManagerDelegate {
    func authenticationManager(_ authenticationManager:
AGSAuthenticationManager, didReceive challenge: AGSAuthenticationChallenge) {
        // NOTE: Never hardcode login information in a production application.
        This is done solely for the sake of the sample.
        let credentials = AGSCredential(user: "cgis_user", password:
"cgis_user123")
        challenge.continue(with: credentials)
    }
}

```