Add the Weather Company Data and Watson Text to Speech Services to an iOS App

One of the most common services used in Mobile Apps is Geolocation. Knowing the latitude and longitude corresponding to the location of a device, can be the input to other useful services such as Weather Forecasting; and adding "speaking" capabilities to an App is a simple, but powerful way to innovate.

Following this guide, you will learn how to use the Weather Company Data and Watson Text to Speech services in Bluemix, while adding the corresponding capabilities to an iOS App that shows the current address where the device is located, according to its geographic coordinates.
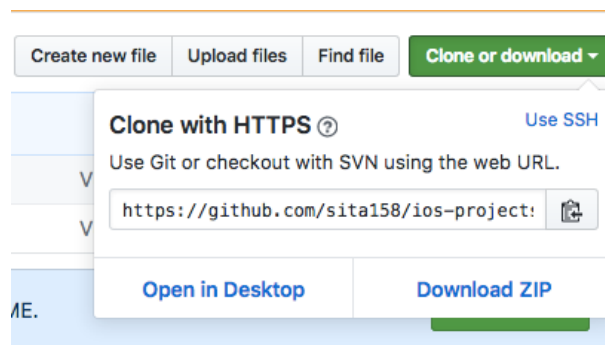
As pre-requisites you will need:

- Xcode
- A valid Bluemix account

# Getting the code and testing the version 0 of the App

To get the base code for this exercise, go the following GitHub repository:

https://github.com/sita158/ios-projects-GeoLocationApp

Click on the "Clone or Download" button and get the project in a Zip file:
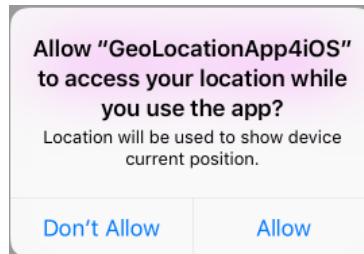

Unzip the downloaded file and right click the GeoLocationApp4iOS.xcodeproj file. Select the "Open with -> Xcode" option:
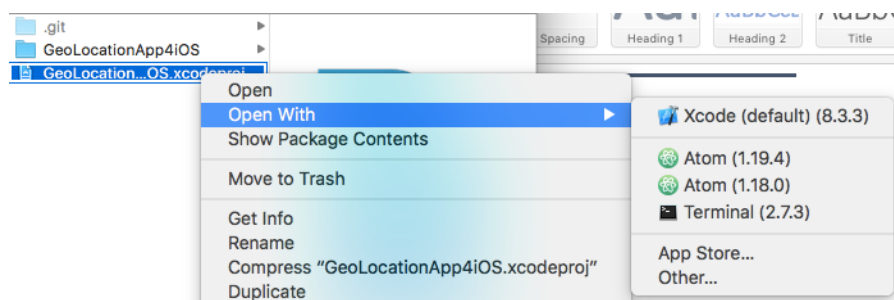
To test the original version of the App, check that a valid simulator is selected as a target and click on the "Build and then run the current scheme" button:
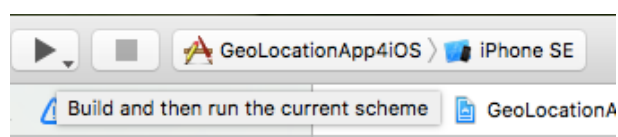
When the app launches, click on "Allow" when asked to provide access to the location services:

The app it will show the address corresponding to the current location of the device. If necessary, click the "Update" button:
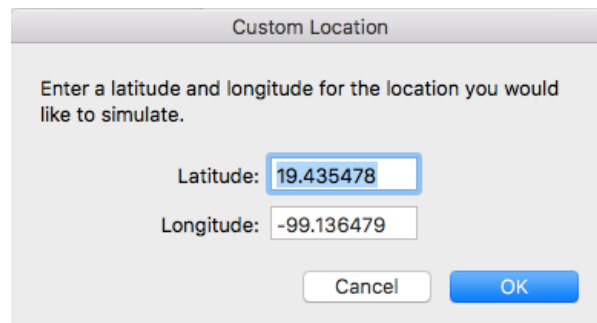

If you are working with a simulator, and want to change the device location, navigate to the "Debug -> Location -> Custom location…" menu, and change the latitude and longitude as desired:

Now you are ready to add the Weather Company Data and the Watson Text to Speech services to the app.

Adding current weather for the location with the Weather Company Data Service

Navigate to bluemix.net in a web browser and Log-in with your valid Bluemix account. In the Services Catalog go to the "Data and Analytics" section and click on the Weather Company Data Service:
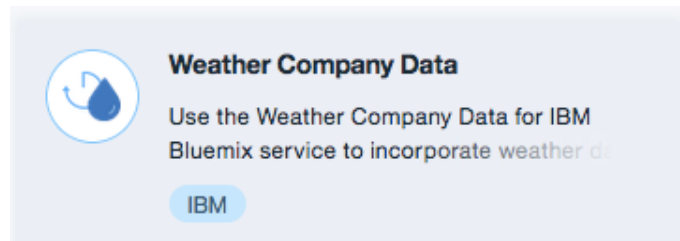
Type a name for the service. Since the name must be unique, use an identificator such as your initials or a number you will remember. Click on "Create":

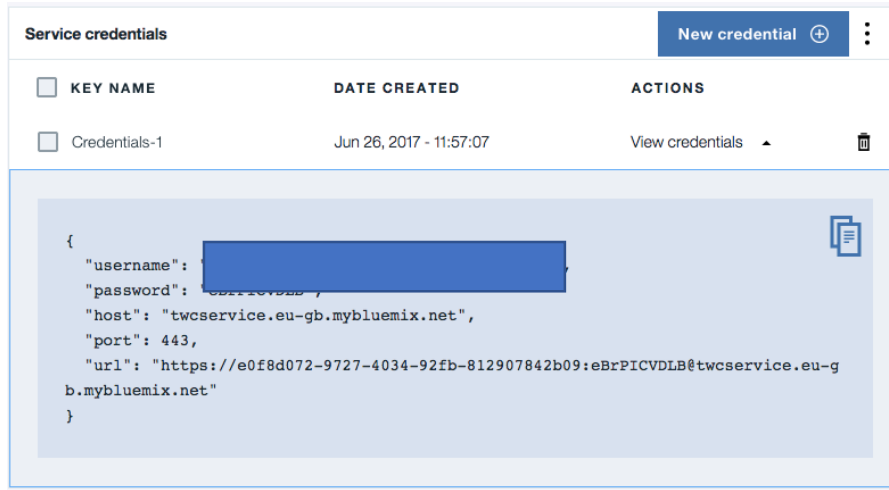Once created navigate to the "Service Credentials" menu, and click on the arrow next to "View Credentials". Keep these values at hand, since they are going to be used in the next steps to code the REST invocation to the Weather Company Data Service:

Now go to Xcode and open the ViewController.swift file under the GeoLocationApp4iOS -> GeoLocationApp4iOS branch of the project tree:
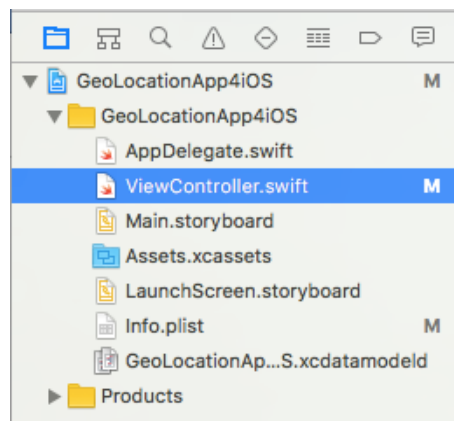
Locate the global variables section and include the following line after the definition of lastLocation:

```
var currentWeatherLabel:UILabel!
```

Locate the viewDidLoad() function and add the following lines between the currentLocationLabel and updateButton definitions:

```
let label2 = UILabel(frame: CGRect(x:0, y:currentLocationLabel.frame.maxY+20, width:
view.frame.width, height:20))
        label2.text = "The weather at that location is:"
        label2.textAlignment = NSTextAlignment.center
        label2.font = UIFont(name: "ArialMT", size: 14)
        view.addSubview(label2)
```

```
        currentWeatherLabel = UILabel(frame: CGRect(x: 0, y: label2.frame.maxY+20, width:
view.frame.width, height: 20))
        currentWeatherLabel.text = "--Weather--"
        currentWeatherLabel.textAlignment = NSTextAlignment.center
        currentWeatherLabel.font = UIFont(name: "ArialMT", size: 14)
        view.addSubview(currentWeatherLabel)
```

Change the Y coordinate of the updateButton frame, and place it after the currentWeatherLabel as follows:

```
let updateButton = UIButton(frame: CGRect(x: 50, y: currentWeatherLabel.frame.maxY+30,
width: view.frame.width-100, height: 30))
```

Incorporate the following two functions in your class. You can paste them after the updateUI() function:

```
func getCurrentWeather(coordinate:CLLocationCoordinate2D){
        let urlString =
"<WeatherCompanyDataServiceURL>/api/weather/v1/geocode/\(coordinate.latitude)/\(coordinate.
longitude)/observations.json"
        if let url = URL(string: urlString){
            var request = URLRequest(url: url)
            request.timeoutInterval = 25
            request.httpMethod = "GET"

            let task = URLSession.shared.dataTask(with: request, completionHandler: { (data
: Data?, response :URLResponse?, error : Error?) in

                if error == nil{
                    let httpResponse = response as! HTTPURLResponse

                    if httpResponse.statusCode == 200 {
                        //OK
                        var dictionary : NSDictionary?

                        do{
                            dictionary = try JSONSerialization.jsonObject(with: data!,
options: JSONSerialization.ReadingOptions.allowFragments) as? NSDictionary
                        }catch{

                        }

                        if let jsonDict = dictionary{
                            print ("jsonDict: \(jsonDict)")
                            let metadataDict = jsonDict["metadata"] as! NSDictionary
                            let code = metadataDict["status_code"] as! NSNumber
                            print("Status code: \(code.intValue)")
                            if code.intValue == 200{
                                let dataDict = jsonDict["observation"] as! NSDictionary
                                let temp = dataDict["temp"] as! NSNumber
                                let wx_phrase = dataDict["wx_phrase"] as! NSString

                                DispatchQueue.main.async {
                                    self.currentWeatherLabel.text="\(temp.intValue)°F,
\(wx_phrase)"
                                }

                            }else{
                                self.showAlert(alertTitle: "Alerta", message: "Error al
accesar el servicio de WCD")
                            }
```

```swift
                }else{
                    self.showAlert(alertTitle: "Alerta", message: "Error en
Serialización")
                }
            }else{
                var httpErrorMessage = ""
                switch httpResponse.statusCode{
                case 404:
                    httpErrorMessage = "Error 404. Servidor no encontrado"
                case 500:
                    httpErrorMessage = "Error 500. Error interno"
                default:
                    httpErrorMessage = "Error desconocido"
                }

                self.showAlert(alertTitle: "Alerta", message: httpErrorMessage)
            }

        }else{
            let iError = error! as NSError

            var nsErrorMessage = ""

            switch iError.code{
            case -1001:
                nsErrorMessage = "Error 1001. Expiró el tiempo de conexión"
            case -1009:
                nsErrorMessage = "Error 1009. Revisa tu conexión a Internet"
            default:
                nsErrorMessage = "Error desconocido"
            }
            self.showAlert(alertTitle: "Alerta", message: nsErrorMessage)
        }
    })
    task.resume()
    }
}

func showAlert (alertTitle: String, message: String){
    DispatchQueue.main.async {
        let alertController = UIAlertController(title: alertTitle, message: message,
preferredStyle: UIAlertControllerStyle.alert)

        alertController.addAction(UIAlertAction(title: "Aceptar", style:
UIAlertActionStyle.default, handler: { (action) in

        }))

        self.present(alertController, animated: true, completion: nil)
    }
}
```

Now change the string <WeatherCompanyDataServiceURL> marked in blue in the above code, and place the URL found in the Service Credentials.

Go to the updateUI() function and insert the following line after the place where the value of self.currentLocationLable.text is set:

```swift
                self.getCurrentWeather(coordinate: coordinate)
```

The code should look like this:

```
if (error == nil){
    if let _places = places {
        if _places.count > 0{
            let place = _places[0]
            print("address: \(place.addressDictionary)")
            if(place.name != nil){    ⚠ String interpolation produces a debug description for an optional value; did you mean to make this explic...
                self.currentLocationLabel.text = place.name!
            }
            self.getCurrentWeather(coordinate: coordinate)
        }else{
            self.currentLocationLabel.text = "Dirección no encontrada"
        }
    } else {
        self.currentLocationLabel.text = "Dirección no encontrada"
    }
}else{
    self.currentLocationLabel.text = "Dirección no encontrada"
}
```

To test the App, check that a valid simulator or physical device is selected as a target and click on the "Build and then run the current scheme" button:
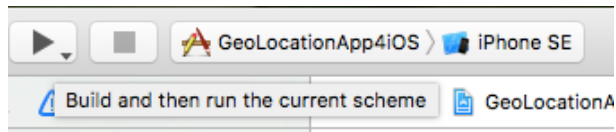

The app it will show the address corresponding to the current location of the device, and the current weather conditions at that location. If necessary, click the "Update" button:
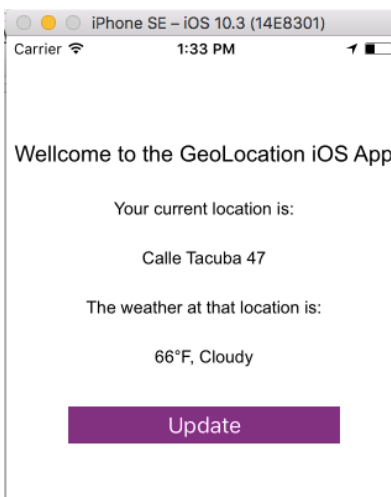
Now you are ready to add the Watson Text to Speech Service to the app.

Adding "speaking" capabilities with the Watson Text to Speech Service

Navigate to bluemix.net in a web browser and Log-in with your valid Bluemix account. In the Services Catalog go to the "Watson" section and click on the Text to Speech service:
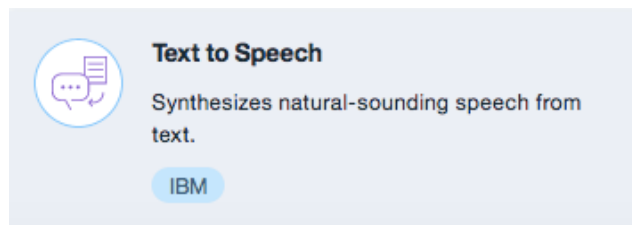
Give the instance of the service a unique name using, for example, your initials, and click the "Create" button:

Once created, navigate on the "Service Credentials" menu and display them by clicking on the arrow next to "View Credentials":

You will use this information later on in the exercise when configuring access to the Text to Speech Service.

To add the Watson Developer Cloud SDK to your project, your will need to install Carthage. Go to a Terminal window and type the following commands:

```
$ brew update
$ brew install carthage
```

Once installed, go to your project folder and create a Cartfile, using the following command:

```
$ touch Cartfile
```

Open the Cartfile and place the following line inside it:

```
github "watson-developer-cloud/swift-sdk"
```

Save the file and run the following command:

```
$ carthage update --platform iOS
```

This will download the Watson Developer Cloud SDK package to the project's folder. Go to <project_folder>/Carthage/Build/iOS and select the TextToSpeechV1.framework folder and select the TextToSpeechV1.framework and RestKit.framework directories:
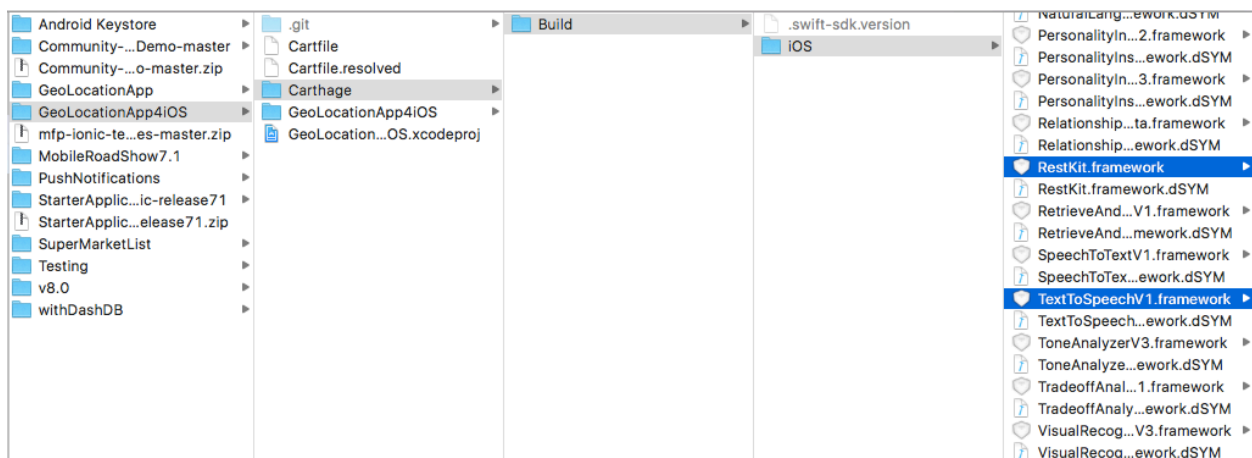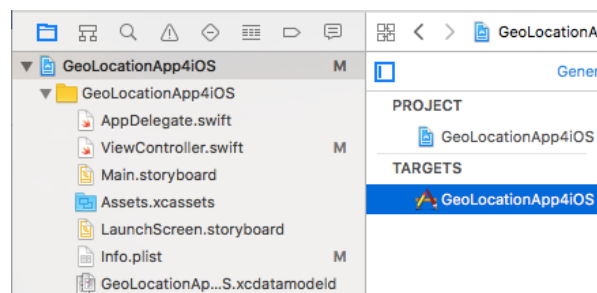
Drag and drop them to the project navigation tree in Xcode. Select the "Copy items if needed" and "Create group" options to import the SDK, and click "Finish":
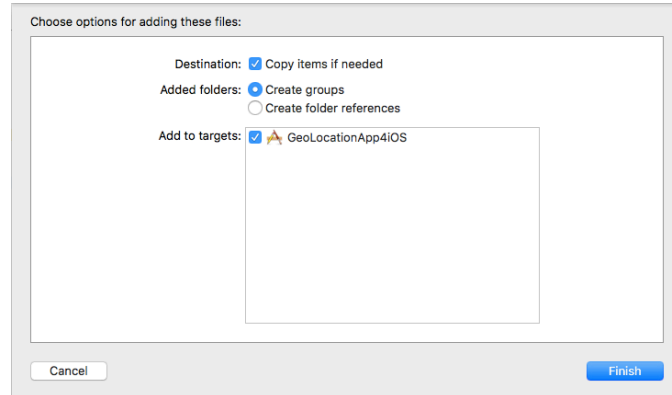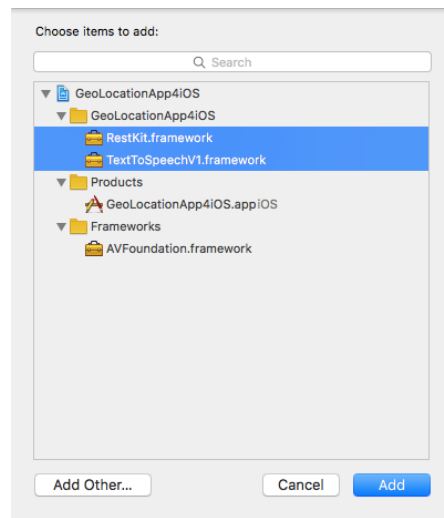

The imported framework should appear in your project's navigation tree:
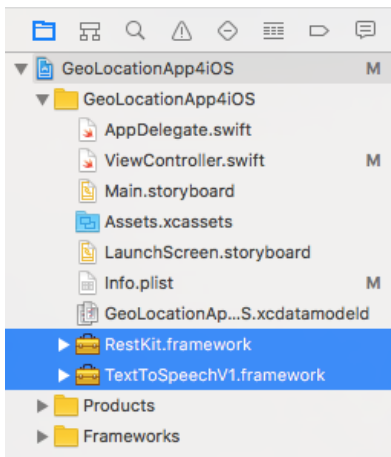
Now click on the GeoLocationApp4iOS root node in the navigation tree of Xcode and select the GeoLocationApp4iOS target:

Select the "General" tab and go to the "Embedded Binaries" section. Click on the plus sign and add the RestKit.framework and TextToSpeechV1.framework:

Open the ViewController.swift file and add the following lines to the import section:

```
import TextToSpeechV1
import AVFoundation
```

Go to the global variables section and add the following definition:

```
var audioPlayer = AVAudioPlayer()
```

In the viewDidLoad() function add a new button after the updateButton definition, using the following code:

```
let readButton = UIButton(frame: CGRect(x: 50, y: updateButton.frame.maxY+30, width:
view.frame.width-100, height: 30))
readButton.setTitle("Read", for: UIControlState.normal)
readButton.setTitleColor(UIColor.white, for: UIControlState.normal)
readButton.backgroundColor = UIColor.purple
readButton.addTarget(self, action: #selector(readAddress), for:
UIControlEvents.touchUpInside)
view.addSubview(readButton)
```

Add the following lines after the getCurrentWeather(coordinate:CLLocationCoordinate2D) function:

```
func readAddress(){
    let username = "<username>"
    let password = "<password>"
    let textToSpeech = TextToSpeech(username: username, password: password)

    let text = self.currentLocationLabel.text
    let failure = { (error: Error) in print(error) }
```

```
        textToSpeech.synthesize(text!, voice: SynthesisVoice.es_Laura.rawValue, failure:
failure) { data in
            self.audioPlayer = try! AVAudioPlayer(data: data)
            self.audioPlayer.prepareToPlay()
            self.audioPlayer.play()
        }
    }
```

Replace the <username> and <password> strings with the values corresponding to your Service Credentials in Bluemix.

Check the following line and change the voice parameter value, according to the type of texts that the Text to Speech Service will be reading:

```
textToSpeech.synthesize(text!, voice: SynthesisVoice.es_Laura.rawValue, failure: failure)
```

The list of available voices in the iOS SDK is the following:

- br_Isabela (portugués)
- de_Birgit (alemán)
- de_Dieter (alemán)
- es_Enrique (español)
- es_Laura (español)
- fr_Renee (francés)
- gb_Kate (ingles británico)

Use the "Build and then run the current scheme" button to test the App on your connected device or simulator. Once you have the last location of the device displayed, click the "Read" button and listen to the Text to Speech service reading the address.
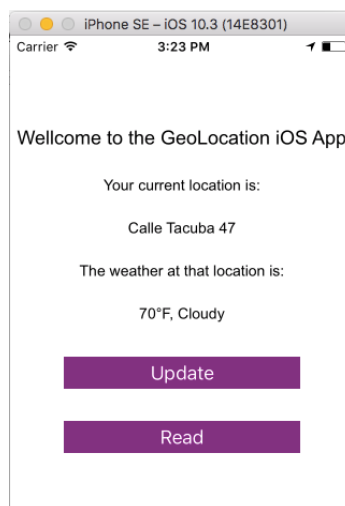
You have successfully completed this recipe!