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:

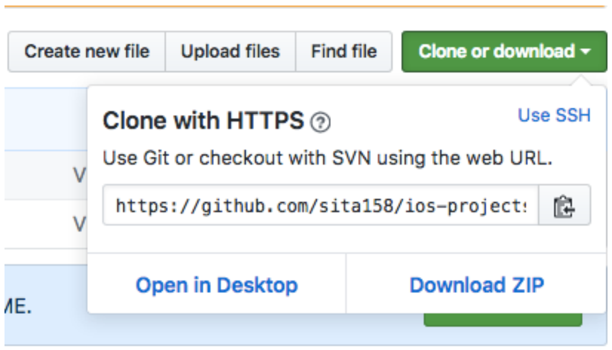
- Xcode
- A valid Bluemix account

Getting the code and testing the version 0 of the App

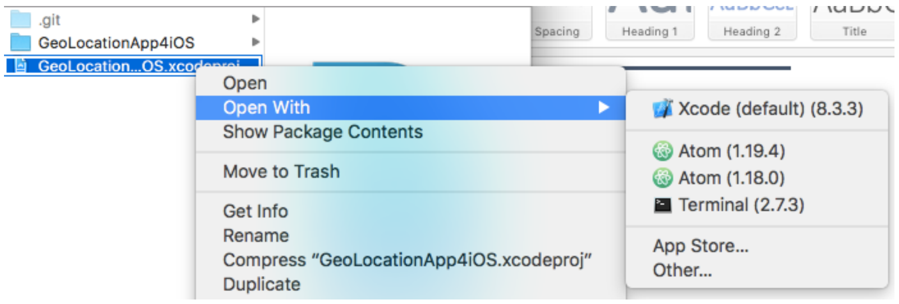
To get the base code for this exercise, go the following GitHub repository:

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

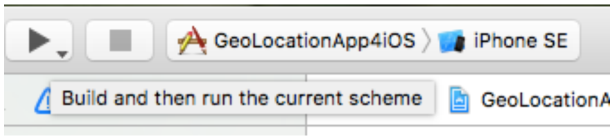
Click on the "Clone or Download" button and get the project in a Zip file:



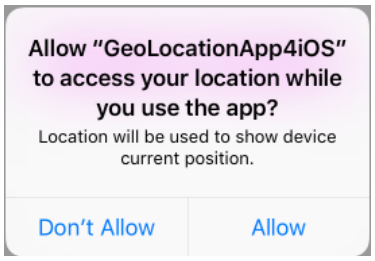
Unzip the downloaded file and right click the GeoLocationApp4iOS.xcodeproj file. Select the "Open with -> Xcode" option:



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:



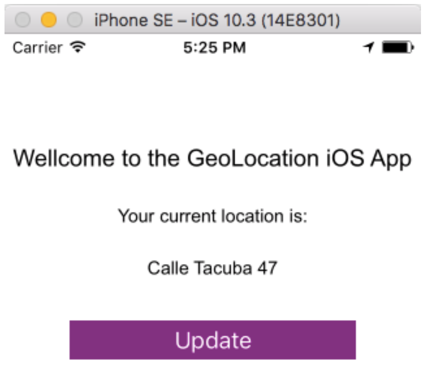
When the app launches, click on “Allow” when asked to provide access to the location services:



Allow “GeoLocationApp4iOS” to access your location while you use the app?

Location will be used to show device current position.

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

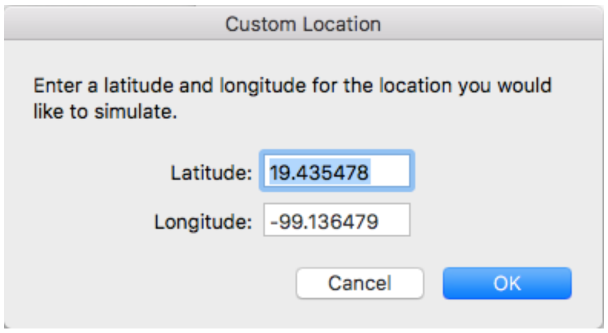


Wellcome to the GeoLocation iOS App

Your current location is:

Calle Tacuba 47

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:



Custom Location

Enter a latitude and longitude for the location you would like to simulate.

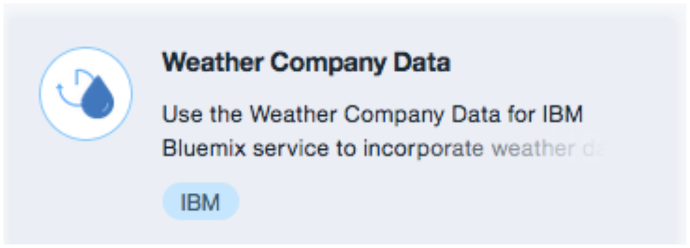
Latitude: 19.435478

Longitude: -99.136479

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

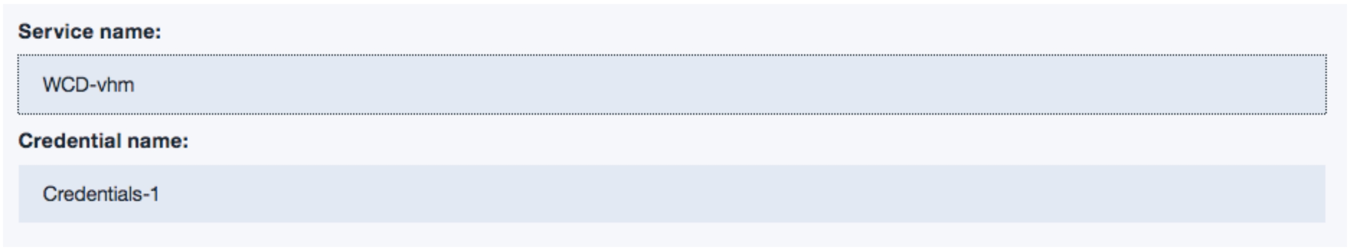
Navigate to bluemix.net in a web browser and Log-in with your valid Bluemix account. In the Services Catalog go to the “Data and Analytics” section and click on the Weather Company Data Service:



Weather Company Data

Use the Weather Company Data for IBM Bluemix service to incorporate weather d

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”:



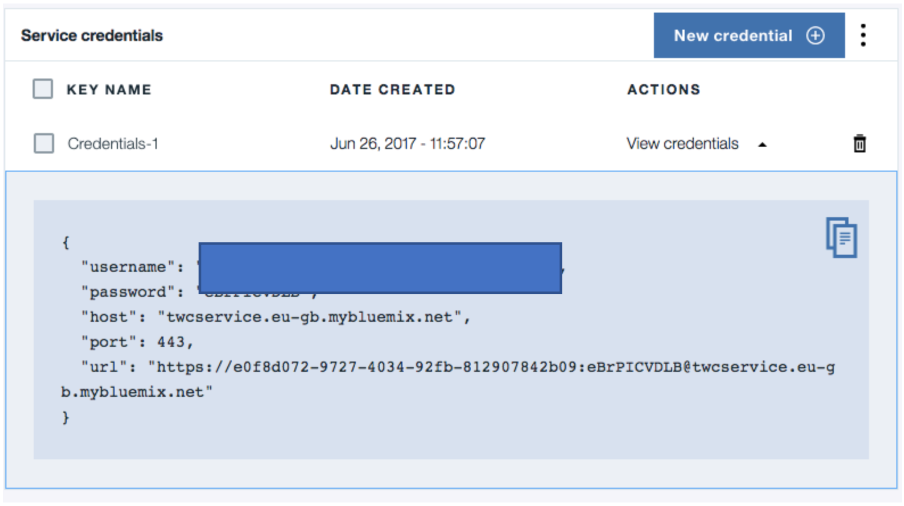
Service name:

WCD-vhm

Credential name:

Credentials-1

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:

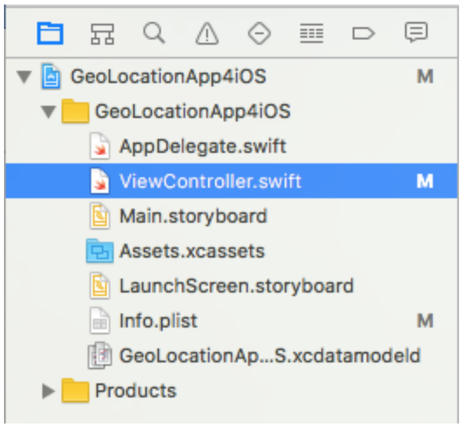


Service credentials | New credential

KEY NAME	DATE CREATED	ACTIONS
Credentials-1	Jun 26, 2017 - 11:57:07	View credentials

```
{
  "username": [redacted]
  "password":
  "host": "twcservice.eu-gb.mybluemix.net",
  "port": 443,
  "url": "https://e0f8d072-9727-4034-92fb-812907842b09:eBrPICVDLB@twcservice.eu-g
b.mybluemix.net"
}
```

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 (you can find the lines of code to be added in the snippets/snippets file of the downloaded project):

```
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:

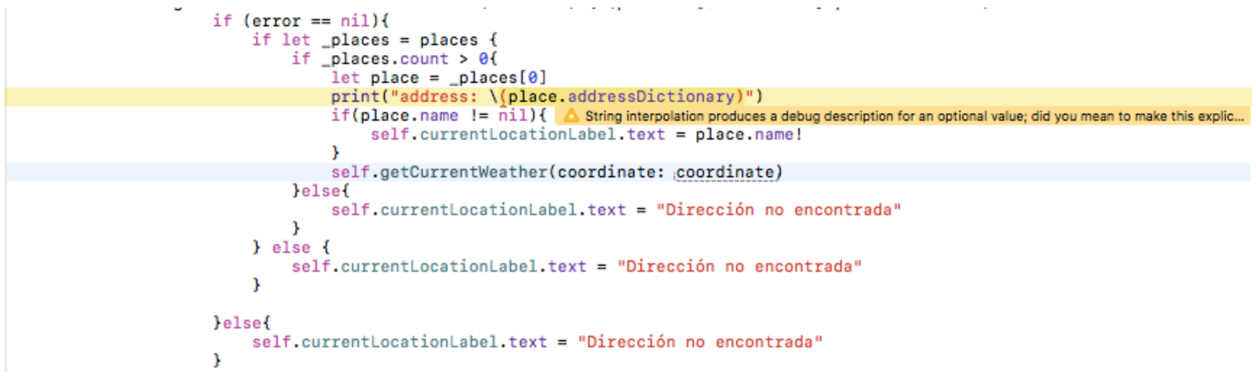
```
func getCurrentWeather(coordinate:CLLocationCoordinate2D){
        let urlString = 
"<WeatherCompanyDataServiceURL>/api/weather/v1/geocode/\(coordinate.latitude)/\(coordinate.
longitude)/observations.json"
        if let url = URL(string: urlString){
            var request = URLRequest(url: url)
            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")
                            }
                        
                        }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.currentLocationLabel.text is set:

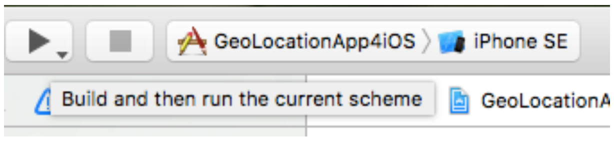
```
self.getCurrentWeather(coordinate: coordinate)
```

The code should look like this:

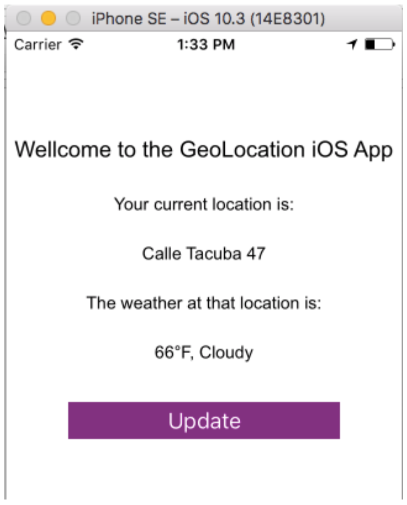


```
if (error == nil){
    if let _places = places {
        if _places.count > 0{
            let place = _places[0]
            print("address: \(place.addressDictionary)")
            if(place.name != nil){  ⚠ String interpolation produces a debug description for an optional value; did you mean to 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:



iPhone SE – iOS 10.3 (14E8301)

Carrier 1:33 PM

Wellcome to the GeoLocation iOS App

Your current location is:

Calle Tacuba 47

The weather at that location is:

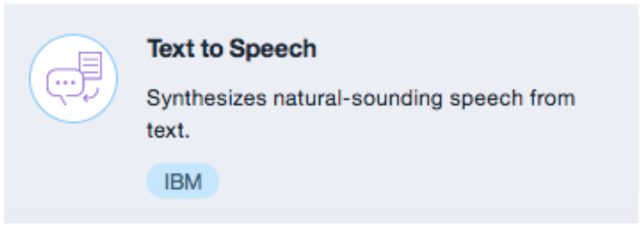
66°F, Cloudy

Update

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:

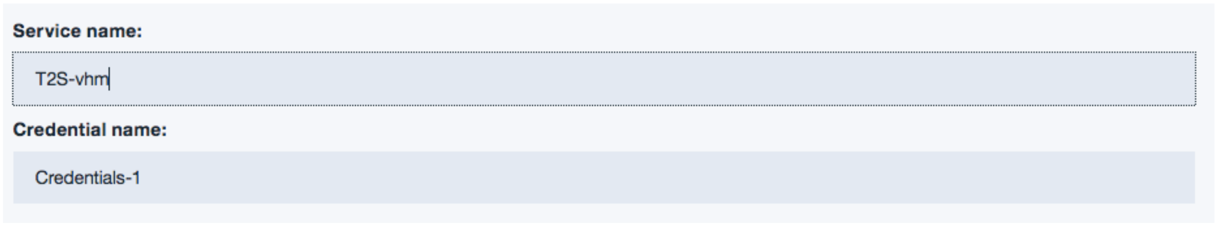


Text to Speech

Synthesizes natural-sounding speech from text.

IBM

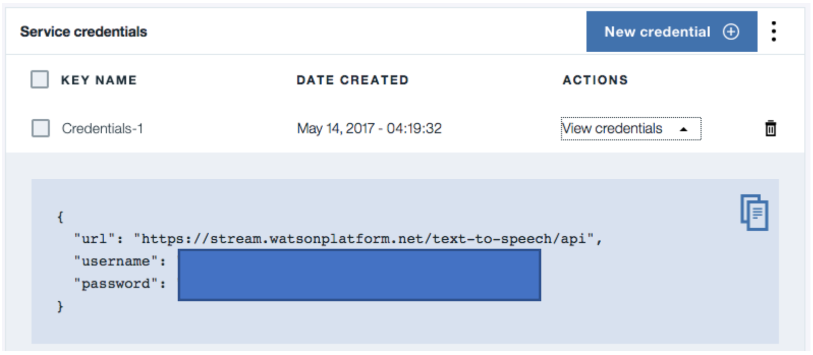
Give the instance of the service a unique name using, for example, your initials, and click the “Create” button:



Service name: T2S-vhm

Credential name: Credentials-1

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



KEY NAME	DATE CREATED	ACTIONS
Credentials-1	May 14, 2017 - 04:19:32	View credentials

```
{
  "url": "https://stream.watsonplatform.net/text-to-speech/api",
  "username": [redacted]
  "password":
}
```

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
```

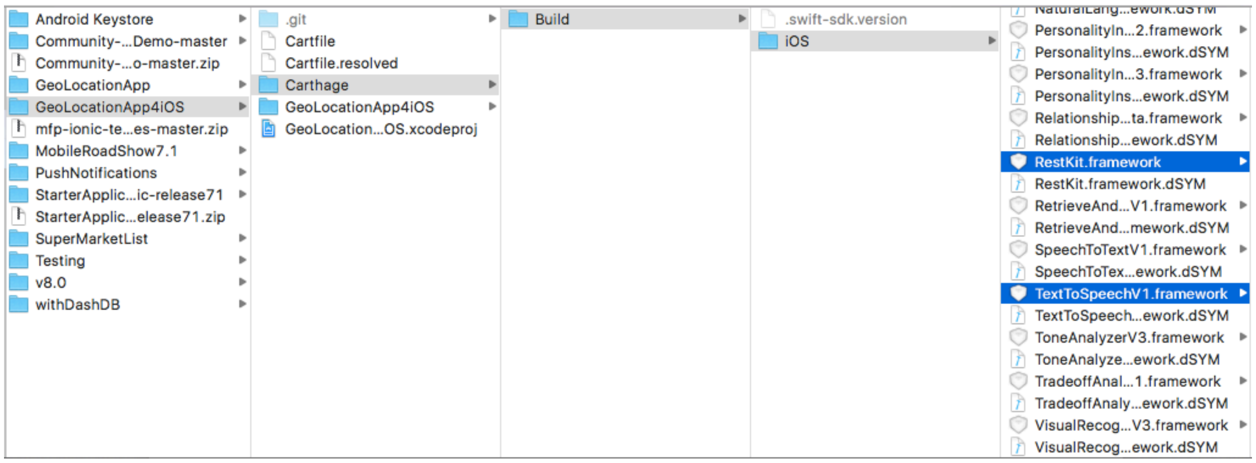
Open the Cartfile and place the following line inside it:

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

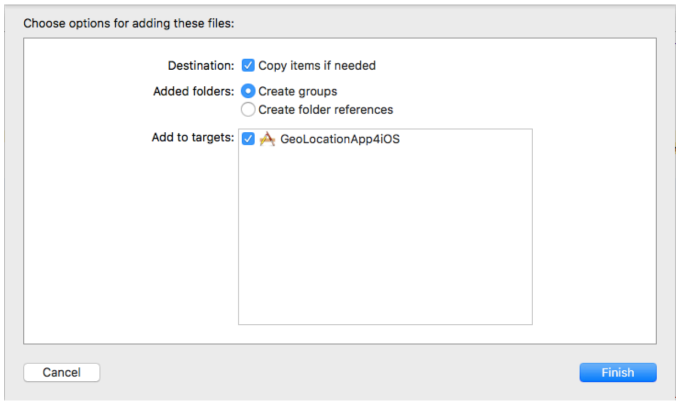
Save the file and run the following command:

```
$ 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:


Android Keystore
Community-...Demo-master
Community-...o-master.zip
GeoLocationApp
GeoLocationApp4iOS
mfp-ionic-te...es-master.zip
MobileRoadShow7.1
PushNotifications
StarterApplic...ic-release71
StarterApplic...elease71.zip
SuperMarketList
Testing
v8.0
withDashDB
.git
Cartfile
Cartfile.resolved
Carthage
GeoLocationApp4iOS
GeoLocation...OS.xcodeproj
Build
.swift-sdk.version
iOS
NaturalLang...ework.dSYM
PersonalityIn...2.framework
PersonalityIns...ework.dSYM
PersonalityIn...3.framework
PersonalityIns...ework.dSYM
Relationship...ta.framework
Relationship...ework.dSYM
RestKit.framework
RestKit.framework.dSYM
RetrieveAnd...V1.framework
RetrieveAnd...mework.dSYM
SpeechToTextV1.framework
SpeechToTex...ework.dSYM
TextToSpeechV1.framework
TextToSpeech...ework.dSYM
ToneAnalyzerV3.framework
ToneAnalyze...ework.dSYM
TradeoffAnal...1.framework
TradeoffAnaly...ework.dSYM
VisualRecog...V3.framework
VisualRecog...ework.dSYM

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”:



Choose options for adding these files:

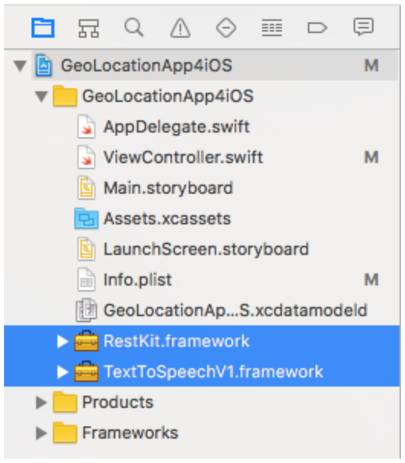
Destination: ☑ Copy items if needed

Added folders: ● Create groups

○ Create folder references

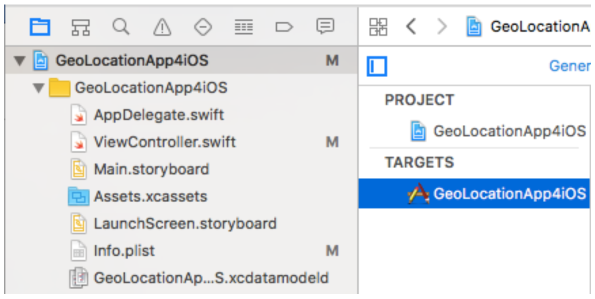
Add to targets: ☑ GeoLocationApp4iOS

The imported framework should appear in your project’s navigation tree:



```
▼ GeoLocationApp4iOS                        M
  ▼ GeoLocationApp4iOS
      AppDelegate.swift
      ViewController.swift                  M
      Main.storyboard
      Assets.xcassets
      LaunchScreen.storyboard
      Info.plist                            M
      GeoLocationAp...S.xcdatamodeld
    ▶ RestKit.framework
    ▶ TextToSpeechV1.framework
  ▶ Products
  ▶ Frameworks
```

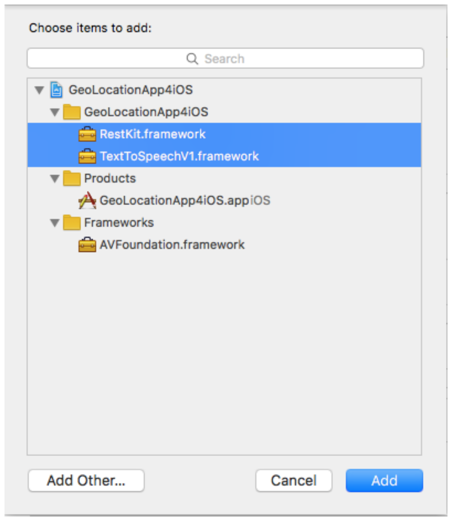
Now click on the GeoLocationApp4iOS root node in the navigation tree of Xcode and select the GeoLocationApp4iOS target:



```
▼ GeoLocationApp4iOS                        M
  ▼ GeoLocationApp4iOS
      AppDelegate.swift
      ViewController.swift                  M
      Main.storyboard
      Assets.xcassets
      LaunchScreen.storyboard
      Info.plist                            M
      GeoLocationAp...S.xcdatamodeld
```

PROJECT
GeoLocationApp4iOS
TARGETS
GeoLocationApp4iOS

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.

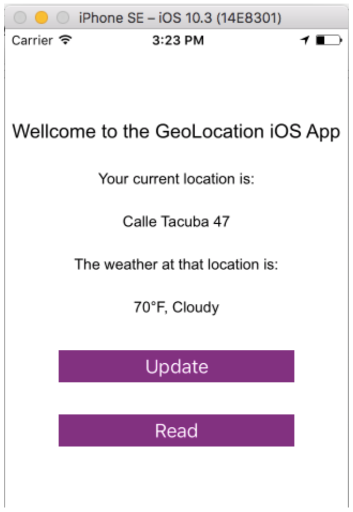
Check the following line and change the voice parameter value, according to the type of texts that the Text to Speech Service will be reading:

```
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.



Wellcome to the GeoLocation iOS App

Your current location is:

Calle Tacuba 47

The weather at that location is:

70°F, Cloudy

You have successfully completed this recipe!