Mobile Apps with the Power of IBM Bluemix

Getting started with the IBM Mobile Application Starters and Bluemix, and integrating IBM Watson services to leverage cognitive computing capabilities.

Download Lab Resources

First, download code from:
https://ibm.biz/BostonMobileLab

This contains complete client and server side code for the project, which only requires minimal configuration. Download the code from the "Download Zip" link and extract it to your working directory on your local machine.

We will also need/use the following tools for interacting wwith the software products:

Cloud Foundry CLI:
https://www.ng.bluemix.net/docs/starters/install_cli.html

CocoaPods:
https://cocoapods.org/

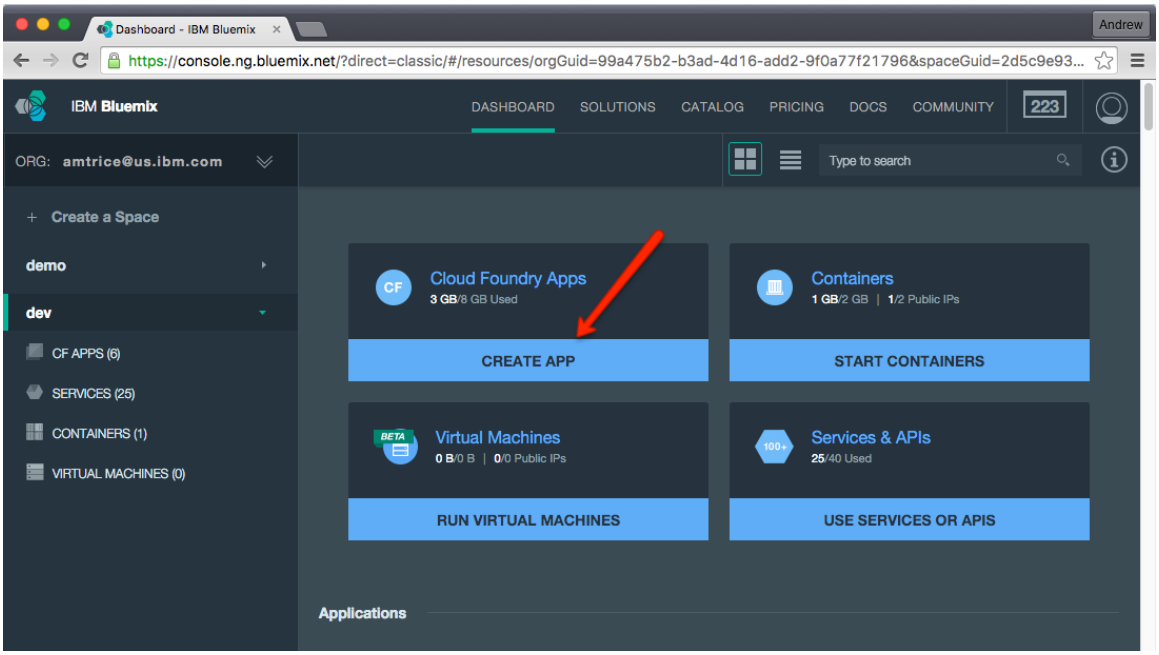
Xcode – requires a Mac

Setup the Bluemix app infrastructure

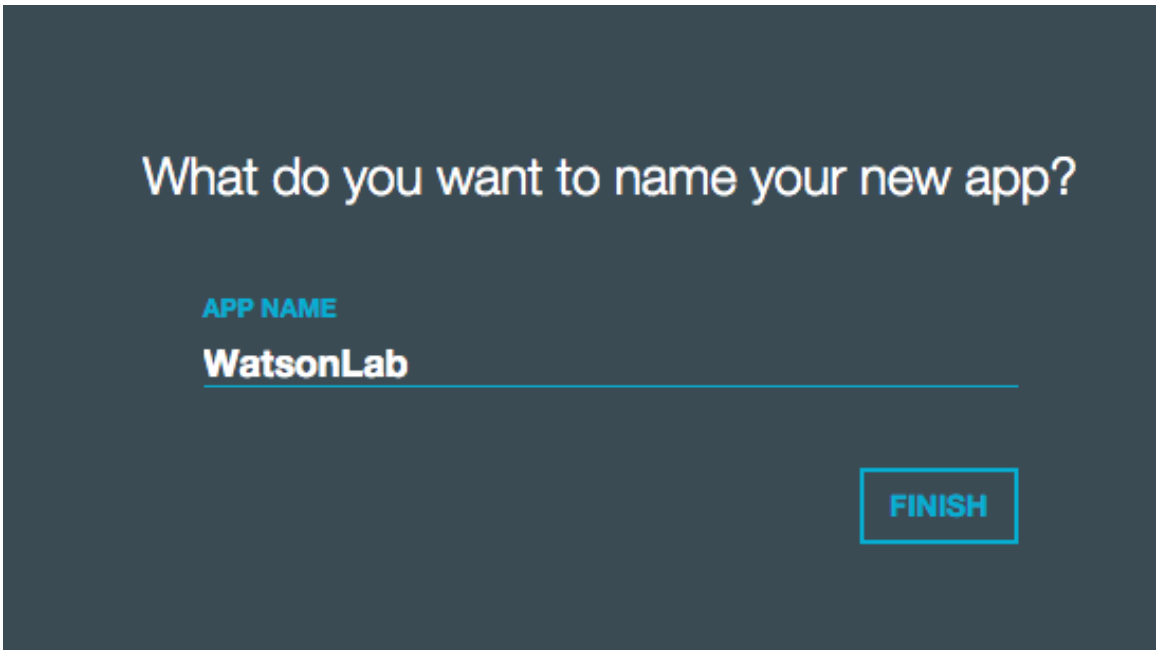
Next we need to setup the Bluemix infrastructure for the app.

1. Log into http://bluemix.net

2. You will be presented with the Bluemix Dashboard. Click on the "Create App" button to walk through the process of creating a new app infrastructure.



3. Enter a name for your app. *This name must be unique across all of Bluemix. Every lab attendee will need a unique app name.*



4. Your mobile app infrastructure has now been created! This includes Mobile Client Access, Cloudant NoSQL Database, Push notifications, and Node.js services.

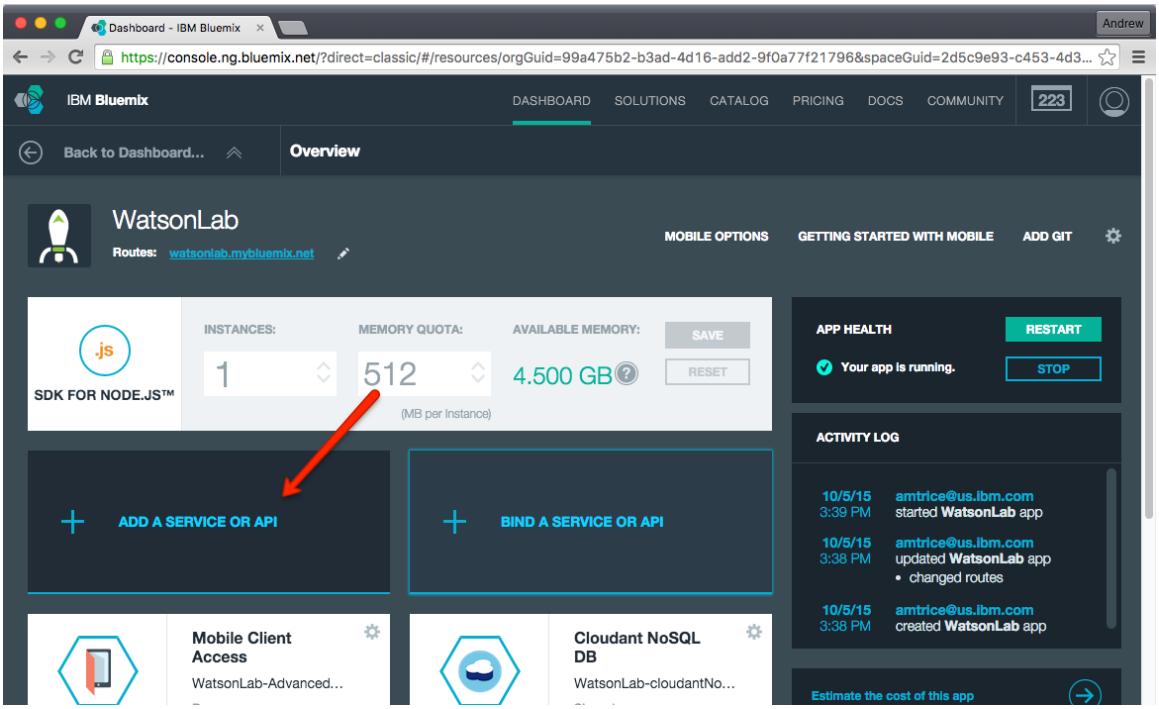
In this sample app, we will only be using Mobile Client access, but everything is already setup if we wanted to integrate Cloudant or Push Notifications.

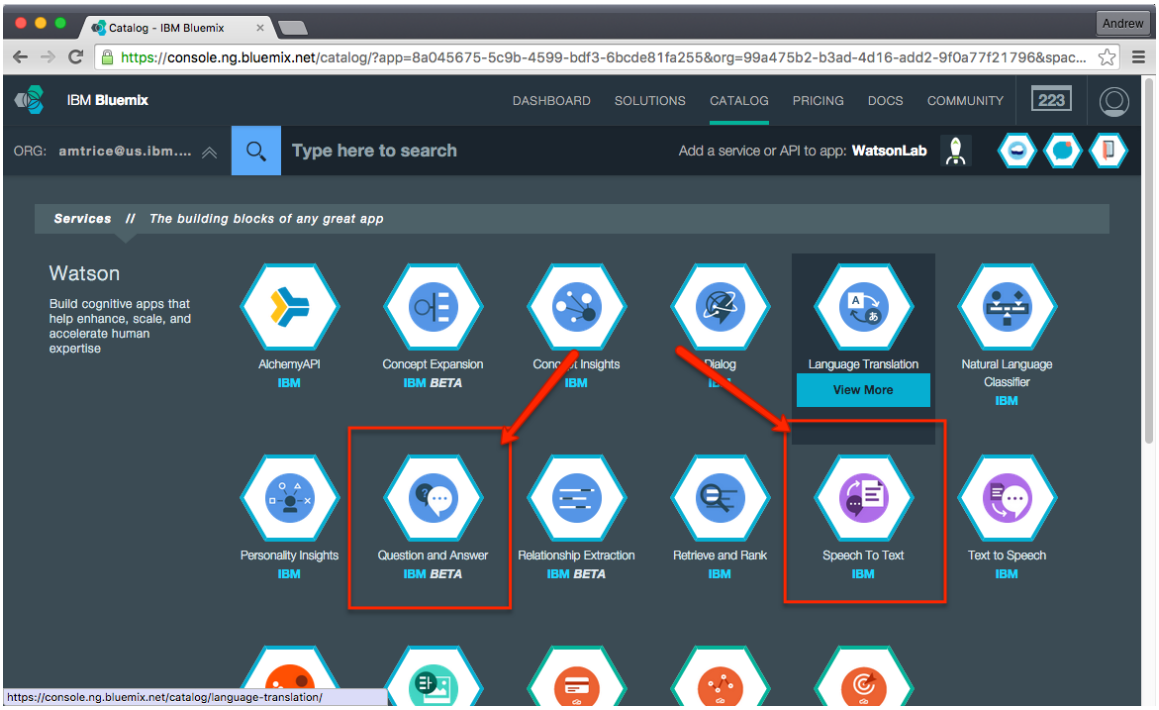
5. Next let's add two Watson services that will be used in our mobile application. Follow these intructions to add the "Watson Question and Answer" and "Speech To Text" services to the Bluemix backend.

    a. From your app's dashboard on Bluemix, select the Add a Service or API" button.

    b. Next, select the "Watson Question and Answer Service" option and walk through all steps to add this to your project using the default settings.

    c. Repeat these steps for the Watson "Speech To Text" service.





Setup Node.js

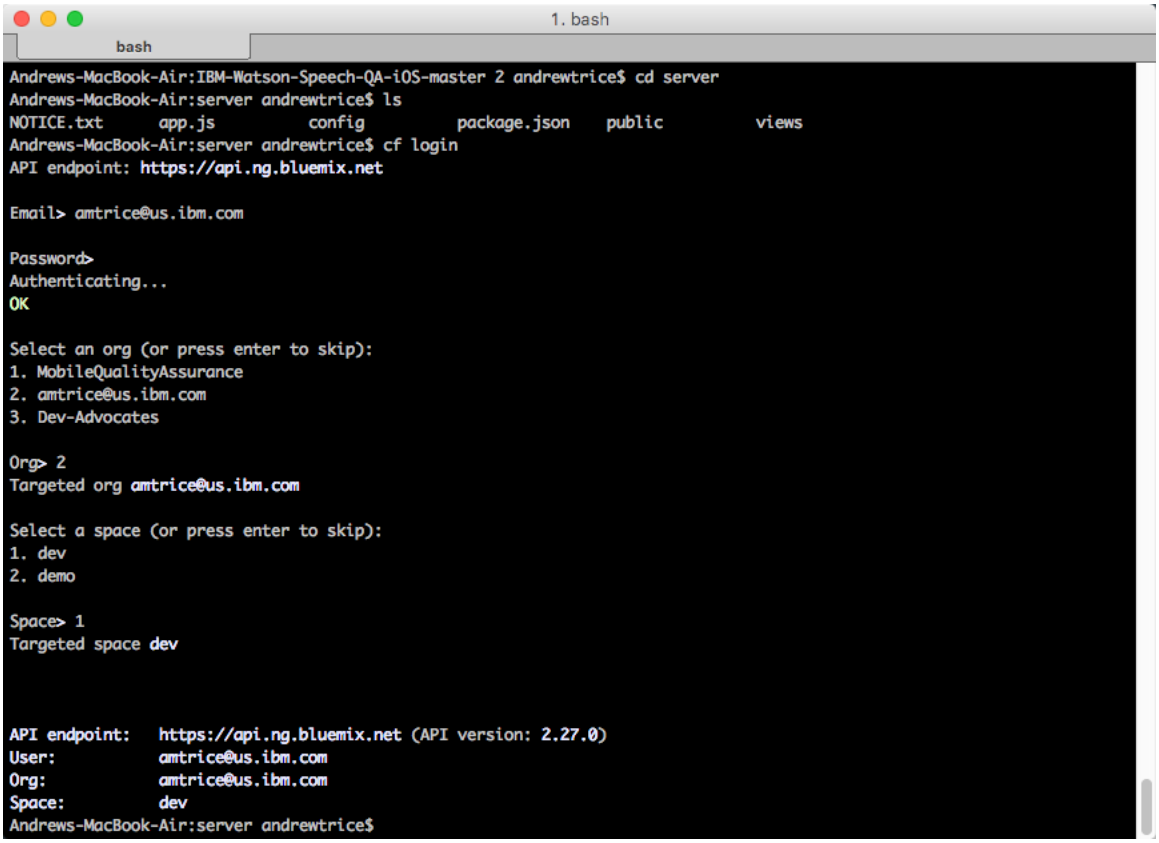
Next we need to setup the backend infrastructure running on Node.js. All off the source code for our application is already complete, we just need to "wire things up". You can browse the source code at any time to see how the pieces work together.

1. Open up an OS X command line terminal.

2. Navigate to the folder containing the source code that was downloaded at the beginning of this lab.

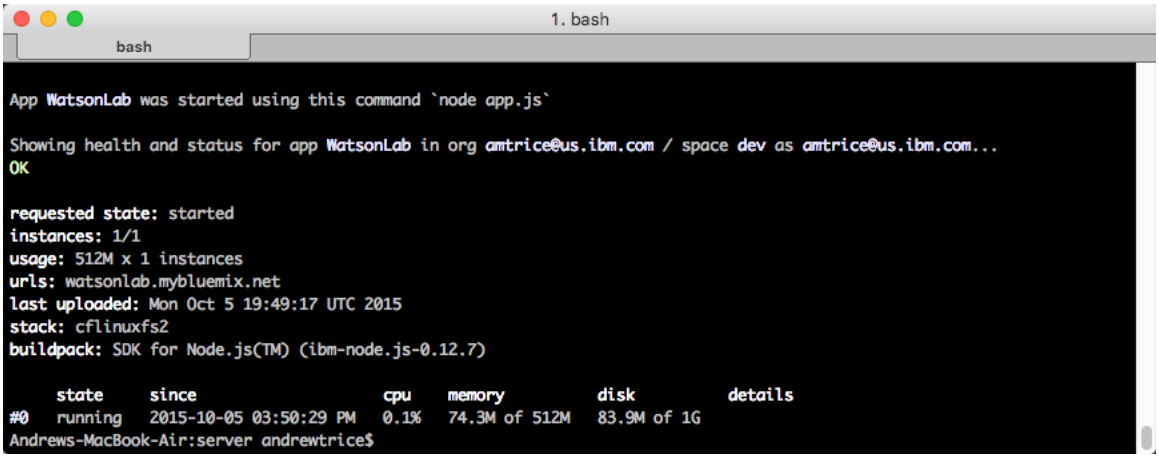
3. Navigate into the "server" directory.

4. Log in using the Cloud Foundry "cf login" command. Follow all prompts to log into the desired space on

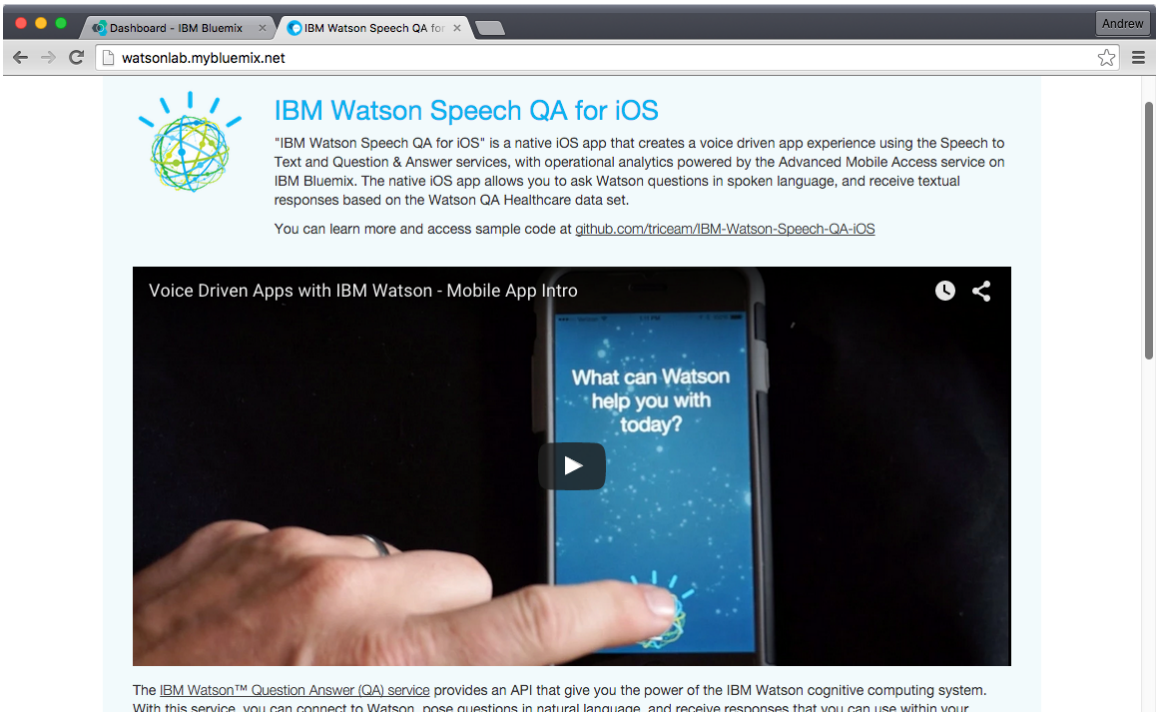


5. Run the "cf push" command with your app's name (the name you used when creating the backend on Bluemix earlier in the lab. In my case this is:

```
cf push WatsonLab
```



6. Now go to your app's URL endpoint in a browser to make sure the sample application has been deployed. This is the URL route shown on your app's dashboard on Bluemix. You will see the web interface shown below, which also allows you to interact with sample questions for the QA service.



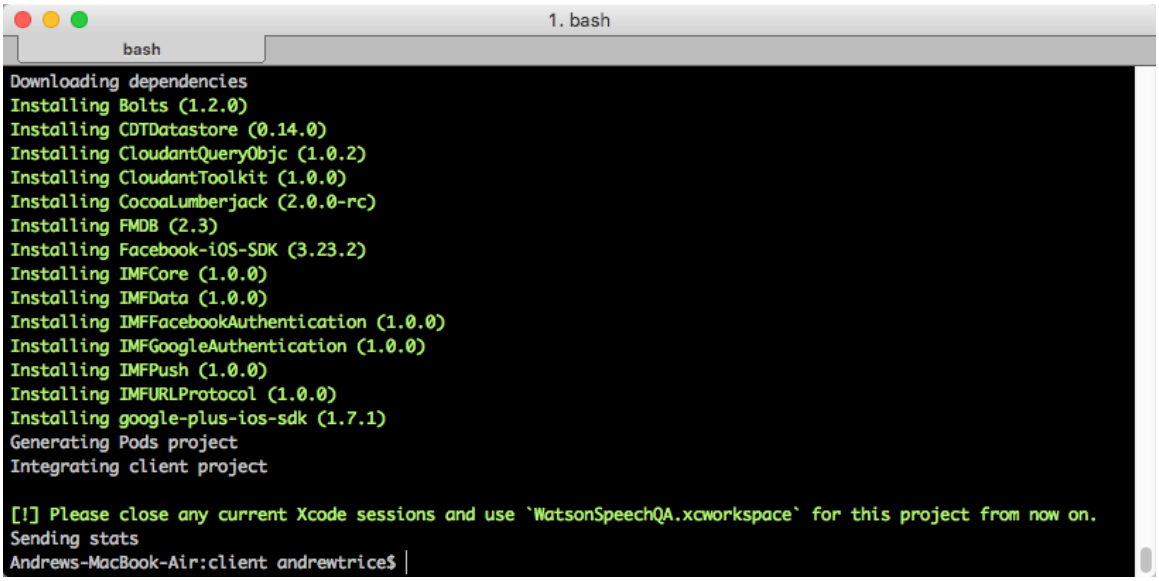
Setup the native iOS project

Next we need to setup the iOS project and configure it to use the Bluemix app infrastructure that we just created.

1. Go back to the command line terminal and navigate to the "client" directory in the downloaded source code

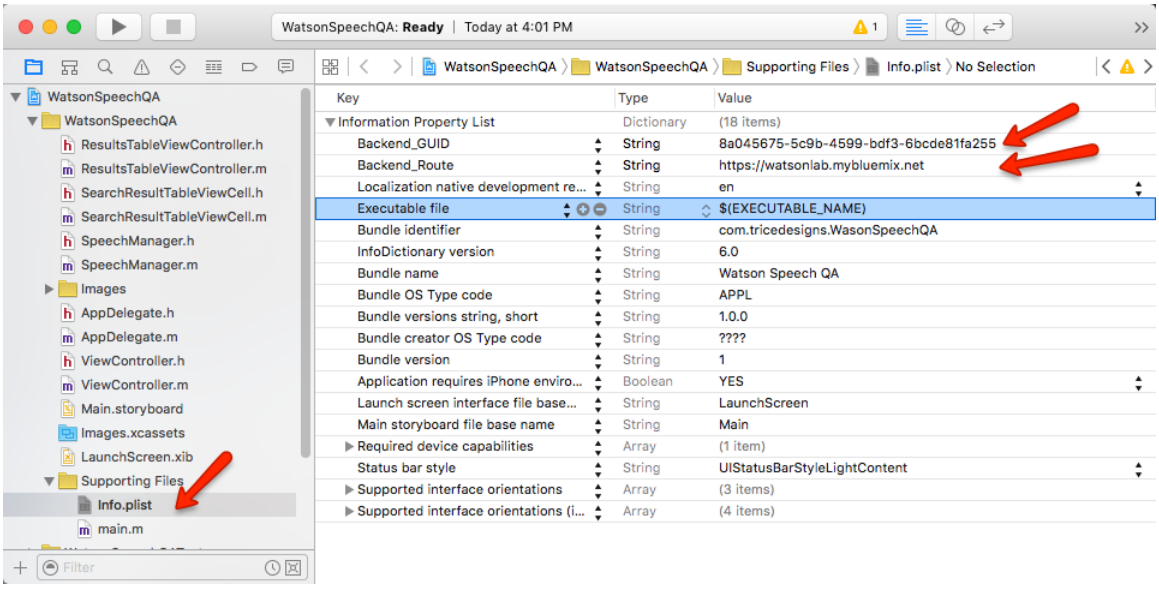
2. Inside the "client" directory, run the command "pod install"

    a. This will pull down all dependency libraries used in the project using the Cocoapods dependency manager. If you do not have Cocoapods, you can download it from http://cocoapods.org



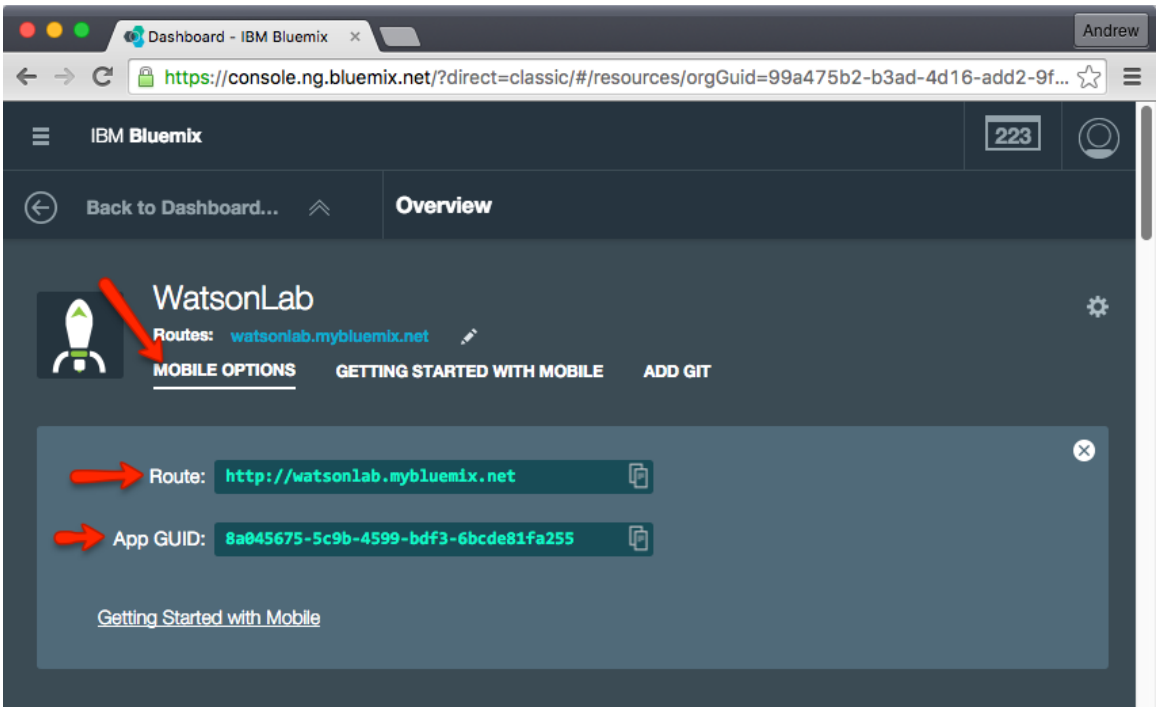
3. Next, open WatsonSpeechQA.xcworkspace in Xcode *(the .workspace file is generated by Cocoapods in the previous step)*

4. In Xcode, open Info.plist and update Backend_GUID and Backend_Route to match the route and app GUID for your app on Bluemix.



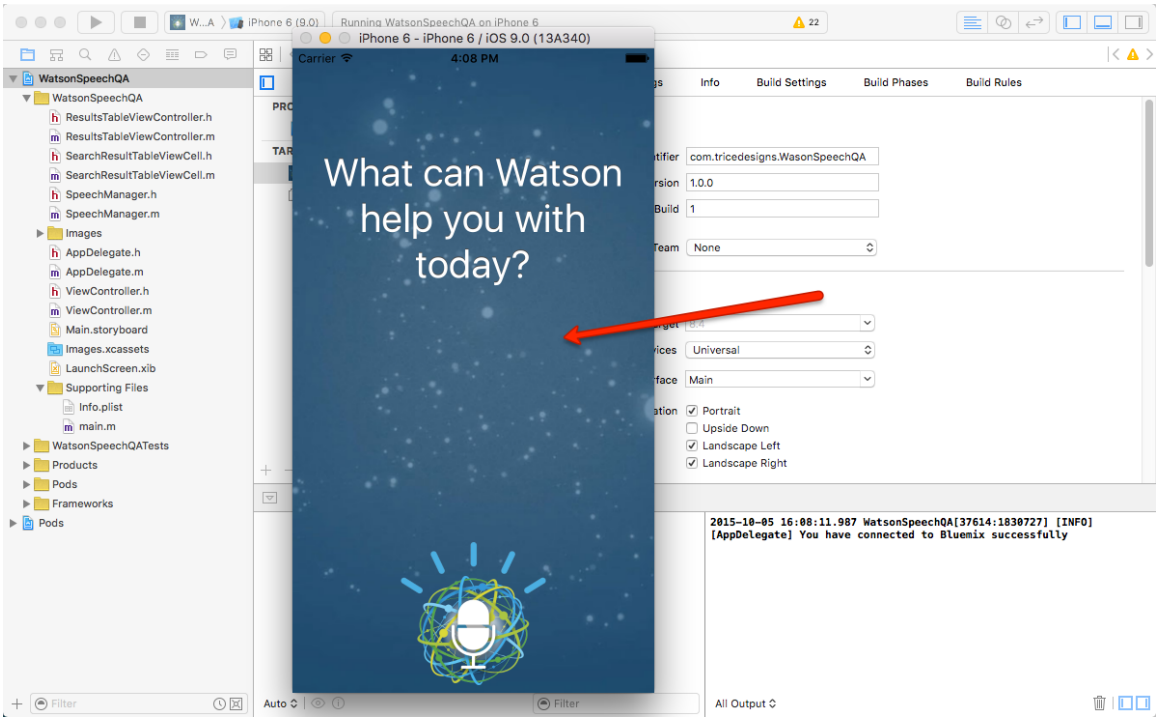
You must use HTTPS! Otherwise iOS 9 Appication Transport Security (ATS) will block the request. To do this, make sure that the value for your Backend_Route is configured to use HTTPS. Bluemix supports both HTTP and HTTPS by default.

To find your app route and app GUID, go to your App on Bluemix and select the “Mobile Otions” link just under the app name and route. The app route and GUID will be shown like the screenshot below.



Next run the app in the Xcode Simulator (or on an iOS device) connected via USB.

The app should now be live and interactive. Just tap the button on the bottom and talk to the app to have Bluemix and Watson start answering your queries. You can speak into your laptop’s mic to use the speech to text service in the Xcode simulator.

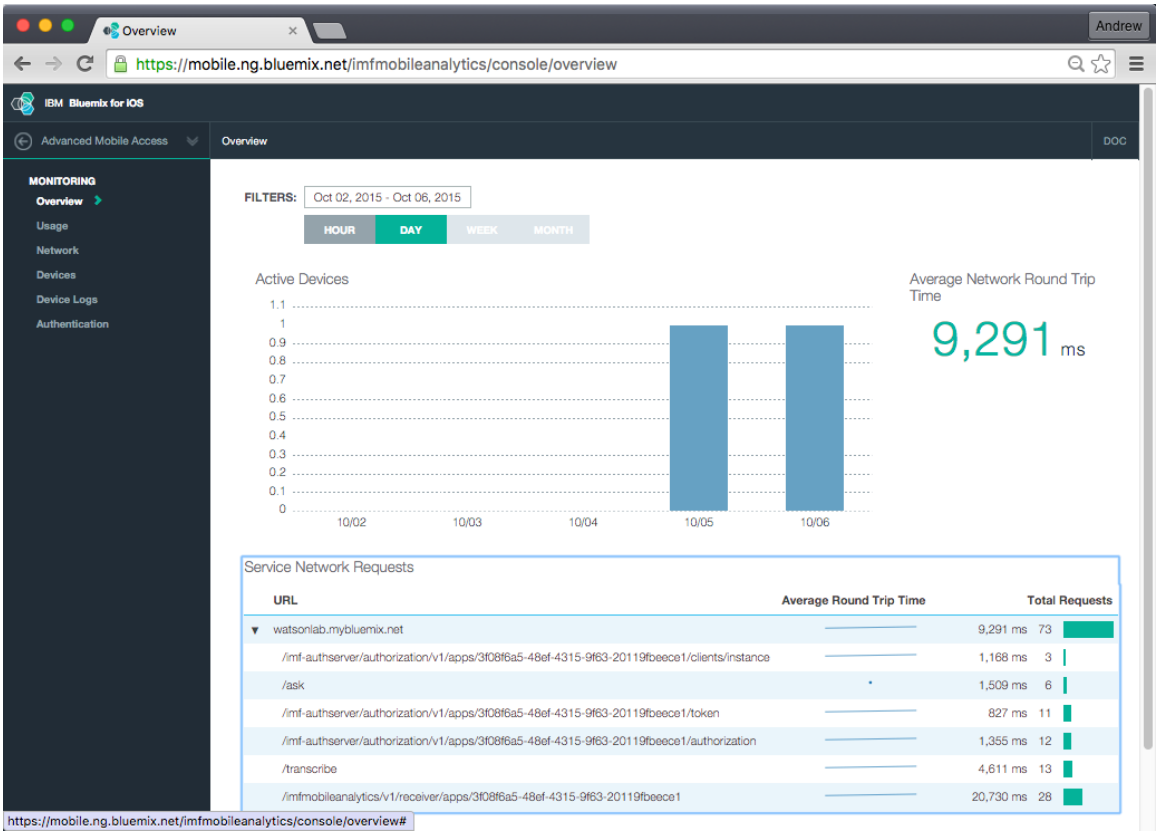


Note: This sample app is calling the Watson Speech To Text service directly via Node.js. To simplify the interaction with Watson Speech To Text and Text To Speech services, and take advantage of lower-latency audio streaming, be sure to check out the new Watson Speech SDK:

1. http://github.com/watson-developer-cloud/speech-ios-sdk
2. http://github.com/watson-developer-cloud/speech-android-sdk

Operational Analytics and Logs

With the Mobile Client Access service, you have access to operational analytics and logs from remote devices "out in the field". To view these, go back to your application dashboard on Bluemix and select "Mobile Client Access". Next go to "Monitoring" on the left side of the screen. Here we can browse through the operational analytics and logs that were collected when running our app.


Overview
Andrew
https://mobile.ng.bluemix.net/imfmobileanalytics/console/overview
IBM Bluemix for iOS
Advanced Mobile Access
Overview
DOC
MONITORING
Overview
Usage
Network
Devices
Device Logs
Authentication
FILTERS:
Oct 02, 2015 - Oct 06, 2015
HOUR
DAY
WEEK
MONTH
Active Devices
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10/02
10/03
10/04
10/05
10/06
Average Network Round Trip Time
9,291 ms
Service Network Requests
URL
Average Round Trip Time
Total Requests
watsonlab.mybluemix.net
9,291 ms 73
/imf-authserver/authorization/v1/apps/3f08f6a5-48ef-4315-9f63-20119fbeece1/clients/instance
1,168 ms 3
/ask
1,509 ms 6
/imf-authserver/authorization/v1/apps/3f08f6a5-48ef-4315-9f63-20119fbeece1/token
827 ms 11
/imf-authserver/authorization/v1/apps/3f08f6a5-48ef-4315-9f63-20119fbeece1/authorization
1,355 ms 12
/transcribe
4,611 ms 13
/imfmobileanalytics/v1/receiver/apps/3f08f6a5-48ef-4315-9f63-20119fbeece1
20,730 ms 28
https://mobile.ng.bluemix.net/imfmobileanalytics/console/overview#