Create a Virtual assistant for iOS with the IBM Cloud Apple Development Console and Watson

Matthew Perrins, IBM Cloud Developer Services, mjperrin@us.ibm.com
Andrew Trice, IBM Cloud Developer Services amtrice@us.ibm.com
Emma Tucker, IBM Cloud Developer Services, etucker@us.ibm.com

Create Virtual assistant for iOS with the IBM Cloud Apple Development Console and Watson

1. Introduction

This lab walks you through the steps required to create a mobile iOS Virtual Assistant application that uses the IBM Watson Assistant service for natural language interaction. You will build and run the components locally with Xcode and interact with IBM Cloud and the Watson Assistant service within the web browser. This process enables developers to become productive with IBM Cloud and the Watson Assistant service rapidly, regardless prior skills and expertise.

The lab is divided into the following main parts:

1. Setup an account with **IBM Cloud**
2. Leverage the **Virtual Assistant for iOS with Watson** starter to create a project and provision an instance of the **Watson Assistant** service
3. Generate app source code, configure your project, and run it locally
4. Modify the conversation dialog intents and structure

Note: *The Watson Conversation service was recently renamed Watson Assistant. Any references to Watson Conversation service are, in fact, also referring to the Watson Assistant service.*

2. Setup

You can setup the prerequisite tools for development on your local machine:

Local Mac Setup:

- Tools for iOS app development
  - XCode IDE 8.0 +
    - https://developer.apple.com/xcode/
  - CocoaPods
    - https://cocoapods.org/
  - Homebrew
    - https://brew.sh/
  - Carthage
    - https://github.com/Carthage/Carthage

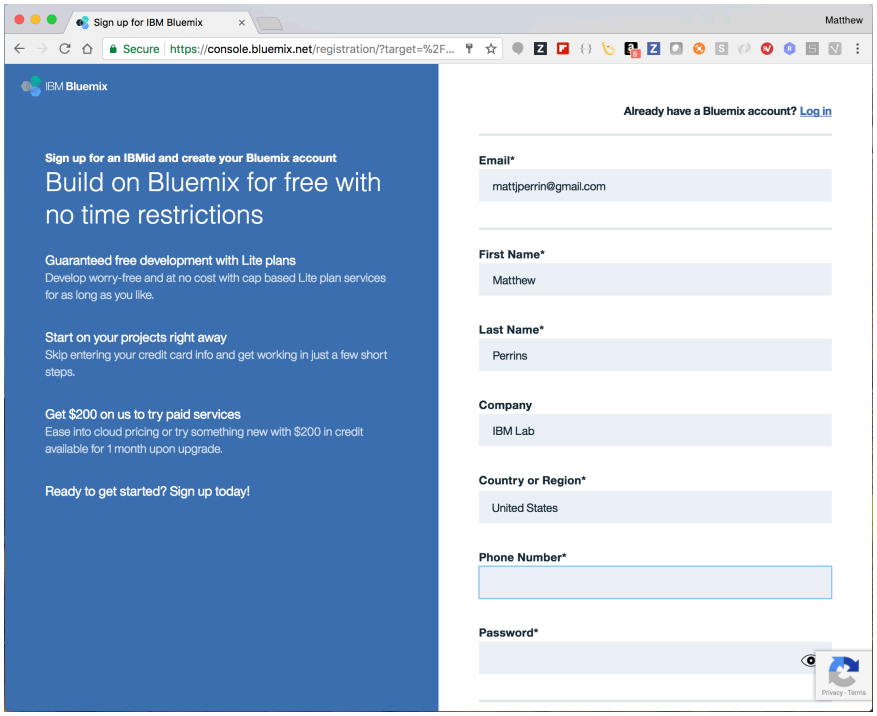
- Full list of requirements
  - https://console.bluemix.net/docs/swift/getting_started/set_up.html#set_up

Setup an IBM Cloud account:

1. To run these lab instructions, you will need to create a trial account on the IBM Cloud. You can do this free of charge and all the instructions contained in this lab do not incur any additional costs.

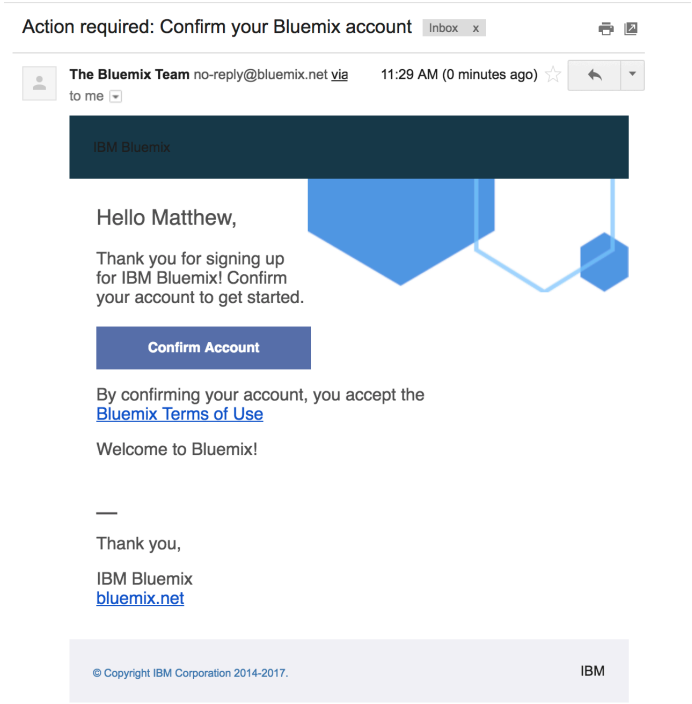
2. Create an IBM Cloud / Bluemix Trial Account by navigating to this link. https://console.bluemix.net, and click on **Create a free Account.**

3. Complete the Registration Form with you details and email address.



4. Click on **Create Account.**

5. Confirm your registration by accessing your email and clicking on **Confirm Account.**



6. You can now log into IBM Cloud.

7. Click on **Login** and enter your email and password.

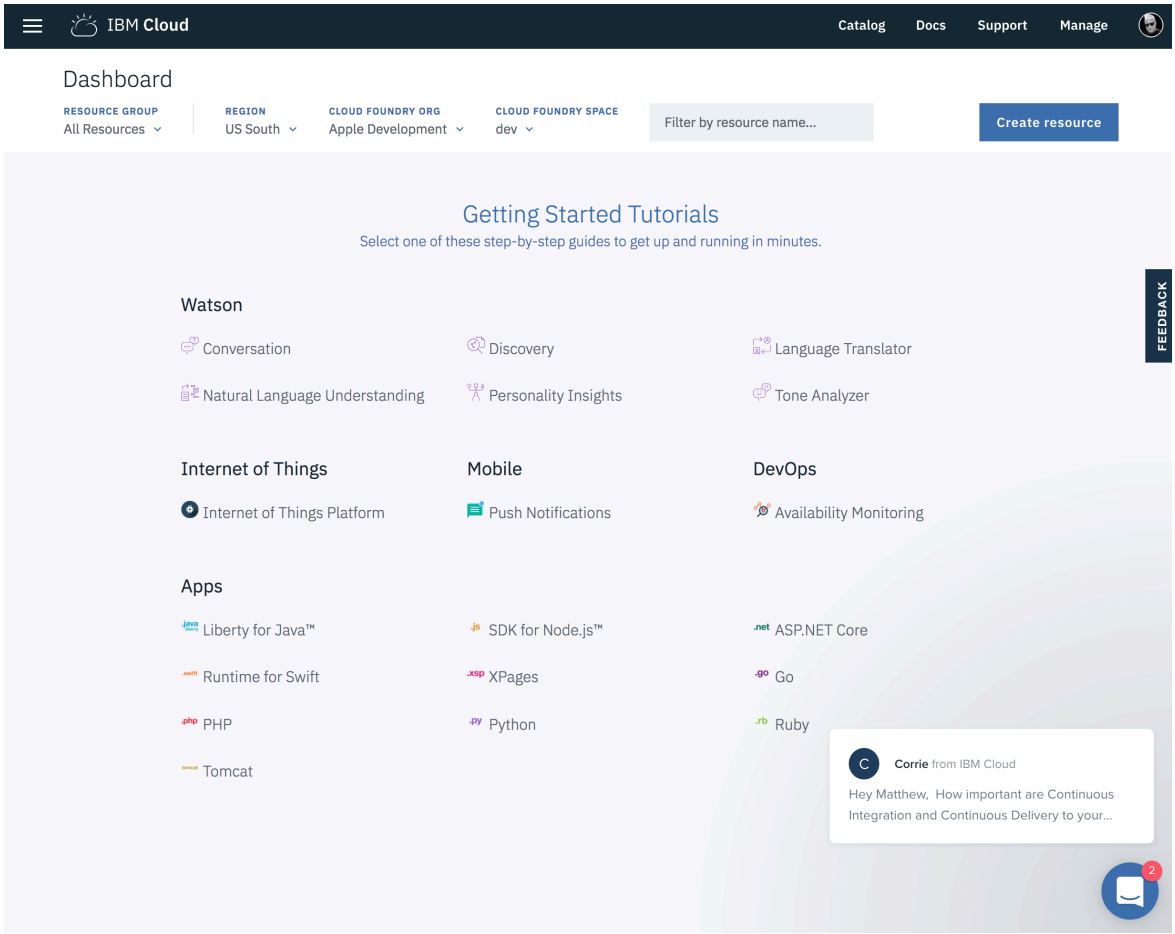
8. You will be asked to create an organization in **US South** region. If not automatically populated, follow these steps to create a dev space:

    a. Enter a name or use **Apple Development.**

    b. Next step creates your space (which is a zone to develop within) create a space called **dev.**

9. Finally click **I'm Ready.**

10. You will finally see the Dashboard View, and this will be empty as you have not created any services or apps at this point.

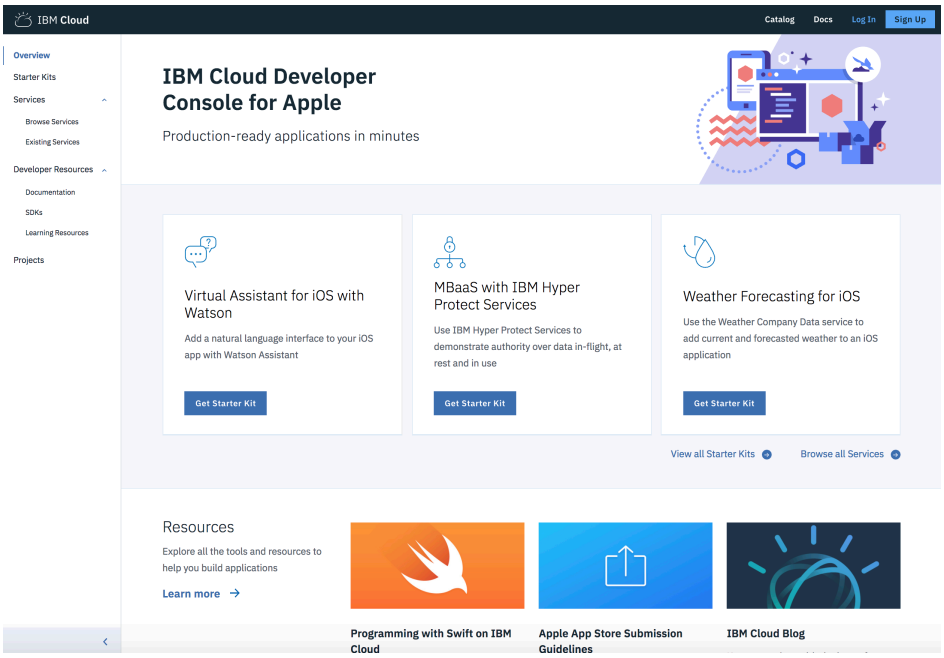


3. Create an iOS Swift App and use the Watson Assistant Service

In this section, you will:

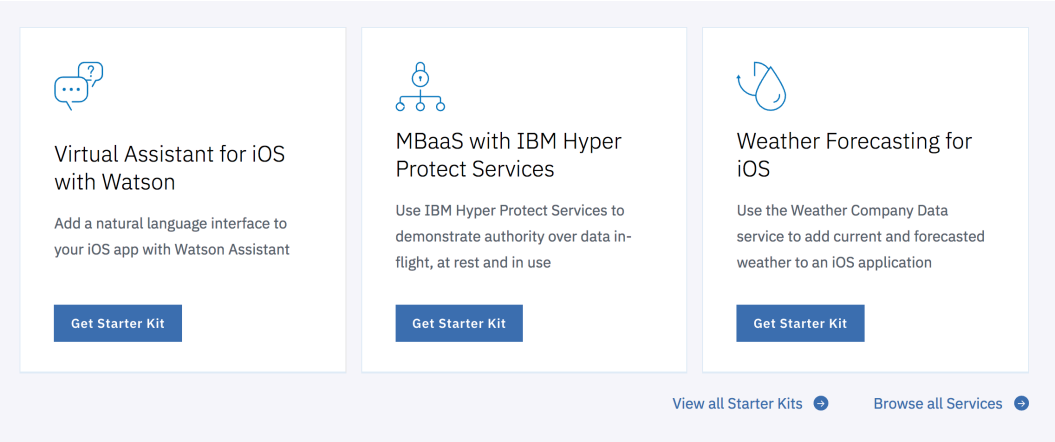
- Familiarize yourself with the **IBM Cloud Developer Console for Apple**
- Create an iOS App using the IBM Cloud Developer Console for Apple using the **Virtual Assistant for iOS with Watson** Starter Kit
- You will download the code and set it up to run in Xcode
- You will test the App in the iOS Simulator

1. Make sure you are logged into the IBM Cloud console. From here we are going to navigate to the **IBM Cloud Developer Console for Apple**. The **IBM Cloud Developer Console for Apple** is focused on helping Apple developers build apps that integrate with the IBM Cloud. Click on **Menu->Apple Development** you should see the Apple Development console overview view, or you can navigate directly to https://console.bluemix.net/developer/appledevelopment

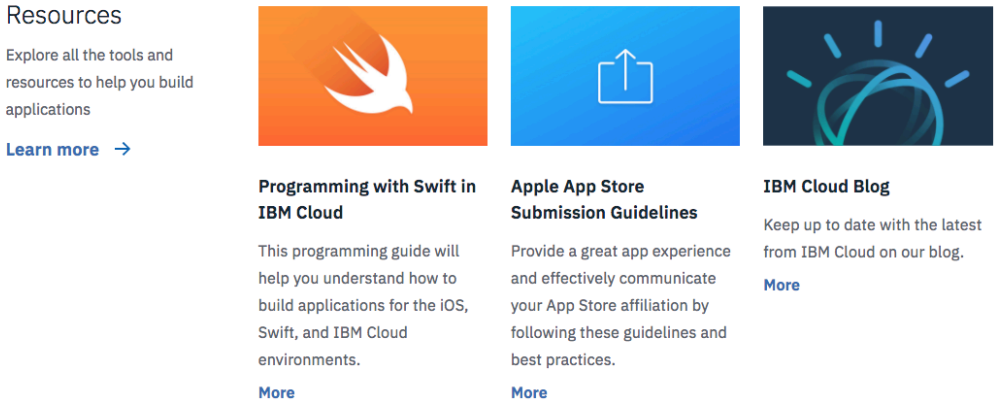


2. Let's first get familiar with the **Overview** screen. The console offers a subset of popular **Starter Kits** on the main Overview page to enable developers to rapidly

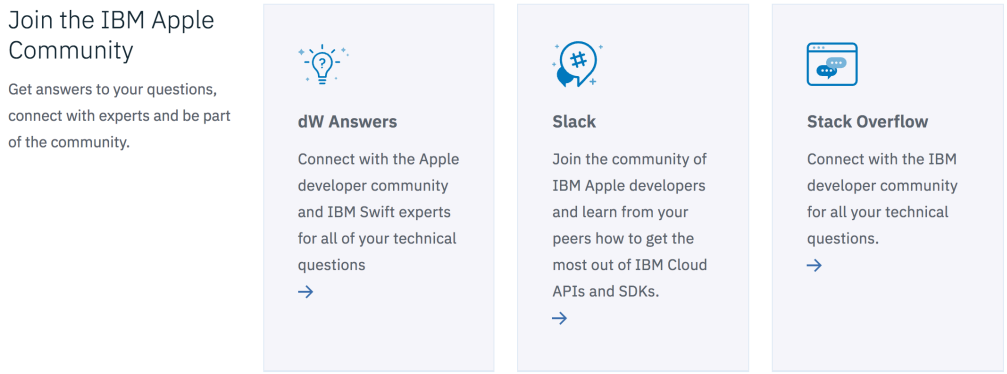
get productive with iOS apps powered by IBM Cloud. There are many more in the Starter Kits menu, but this enables quick discovery.



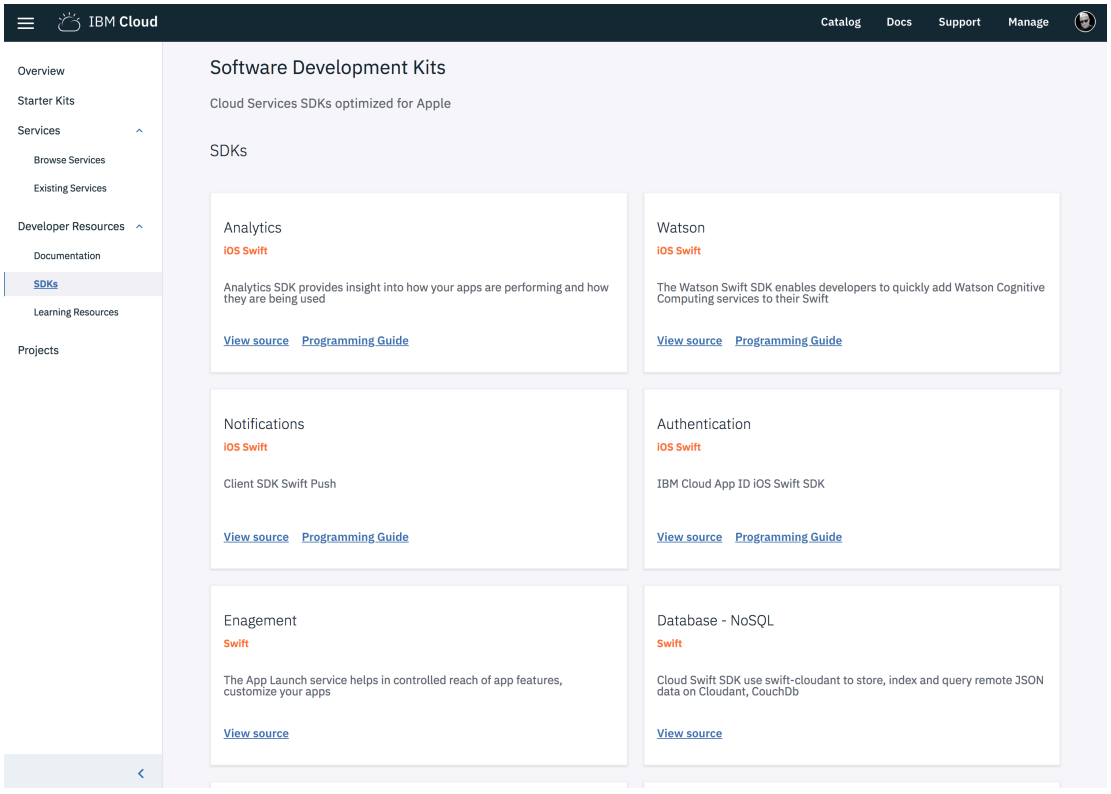
3. Next you'll find the **Resource** section; this is a subset of all the **Learning Resources** that can be found on the left-hand menu. The key resource to spend time understanding is our guide for **Programming with Swift on IBM Cloud**, which contains valuable information for enabling Apple developers to be successful on the IBM Cloud.



4. After **Resources**, you'll find the **Community** area where you can interlock with IBM and Apple engineers to get solutions. You can directly ask questions in Slack or post questions to the wider community in Stack Overflow or use dW Answers for specific IBM technical questions.

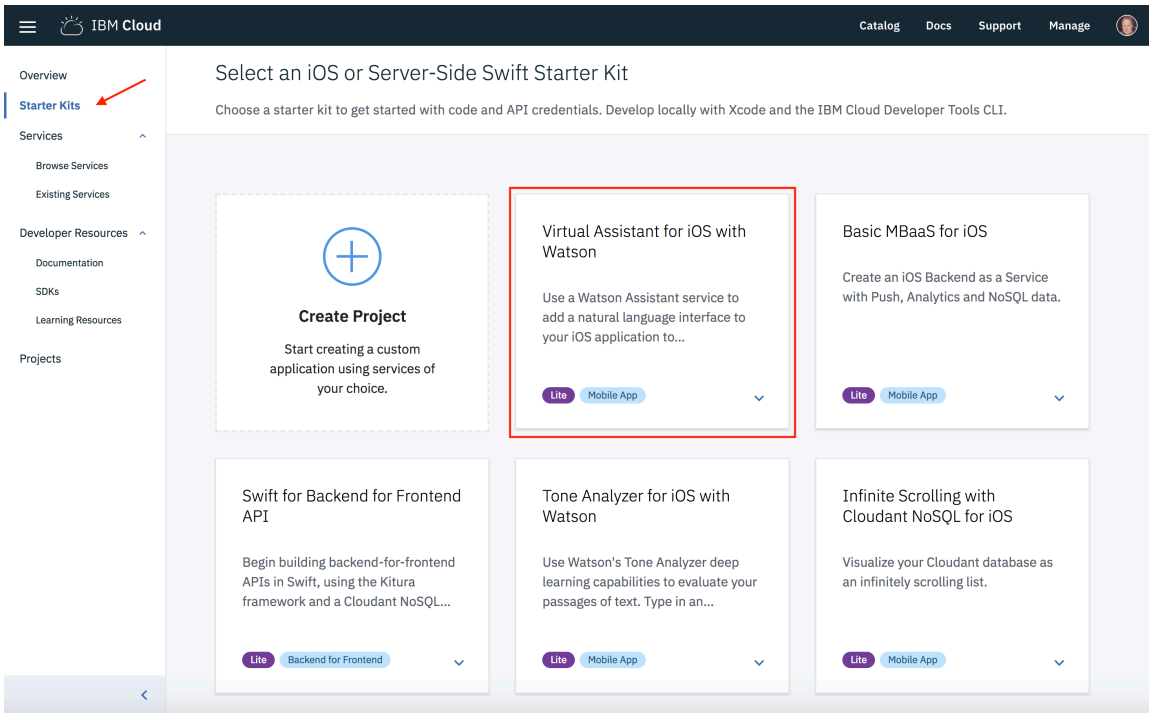


5. If you have existing iOS Apps, you may just want to use a single SDK to help you integrate with the IBM Cloud. The **SDKs** menu will display all the SDKs that are optimized for the Apple development platforms. The SDK tiles also link to the guide for **Programming with Swift on IBM Cloud**. This enables you to get directly to the documentation that supports that SDK. You can then integrate it directly into your existing App.



6. Now let’s start building a conversational iOS App that leverages the Watson Assistant service.

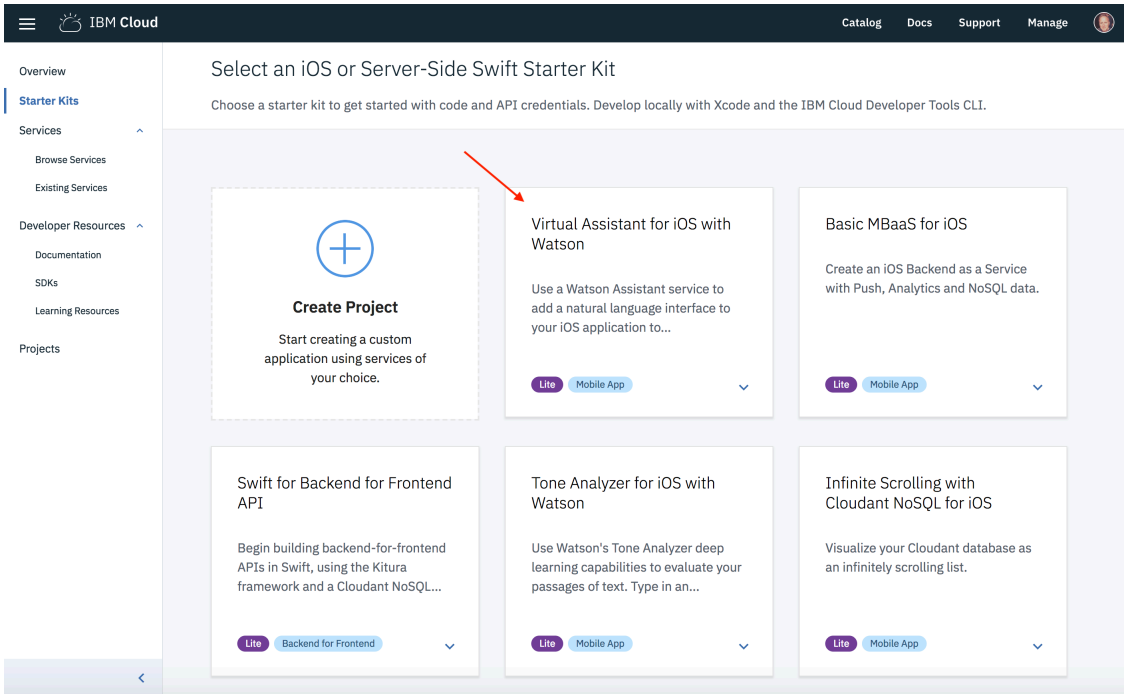
7. Click on **Starter Kits** in the left-hand menu.



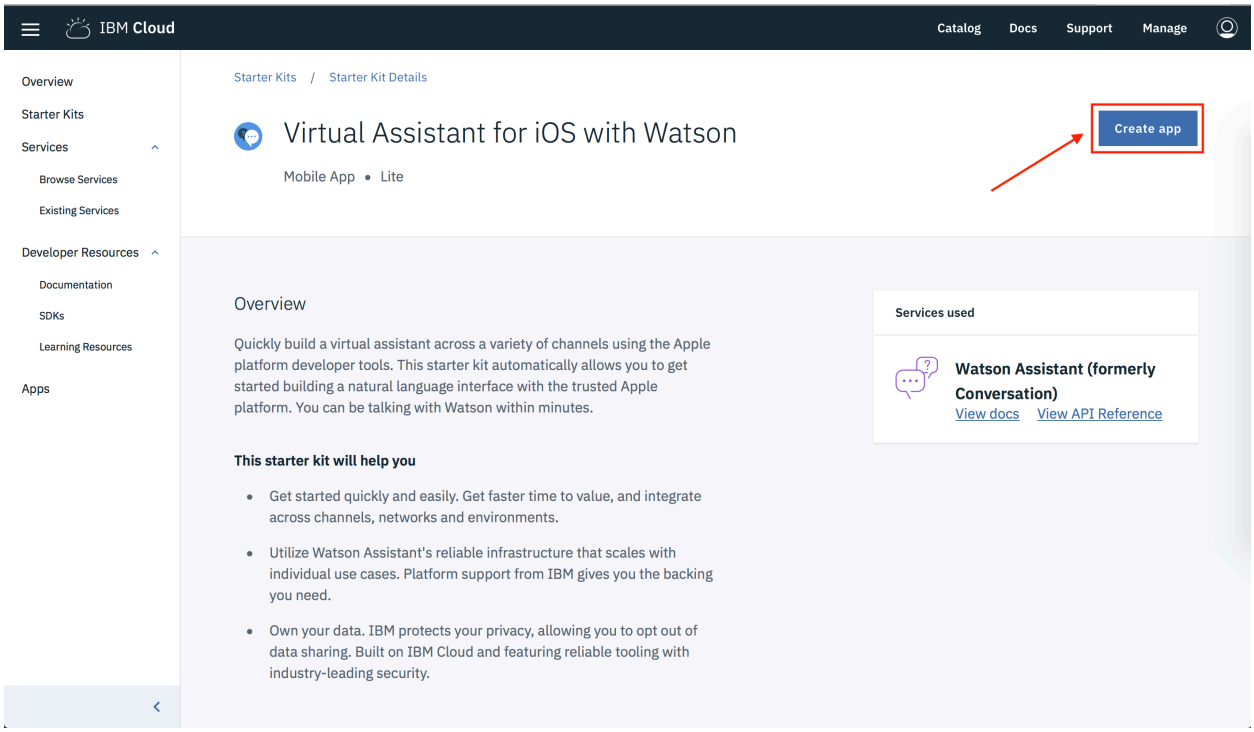
Here you will find a collection of production-ready starter kits that enable you to get up and running very quickly. The key benefit is that you can choose the IBM Cloud Service you want to include for your app and generate an iOS Swift App with just the code you need.

The starter kits cover both client and the server programming models with iOS Swift Apps labeled as **Mobile App** and server-side options labeled as **Backend for Front End** or **Web Apps**, both leveraging the Swift Kitura framework. The final option is the **Create Project** option. This starting point has no use case associated with it and allows you to hand pick the services you'd like to use. It will include the SDKs and SDK initialization logic for just the services you add.

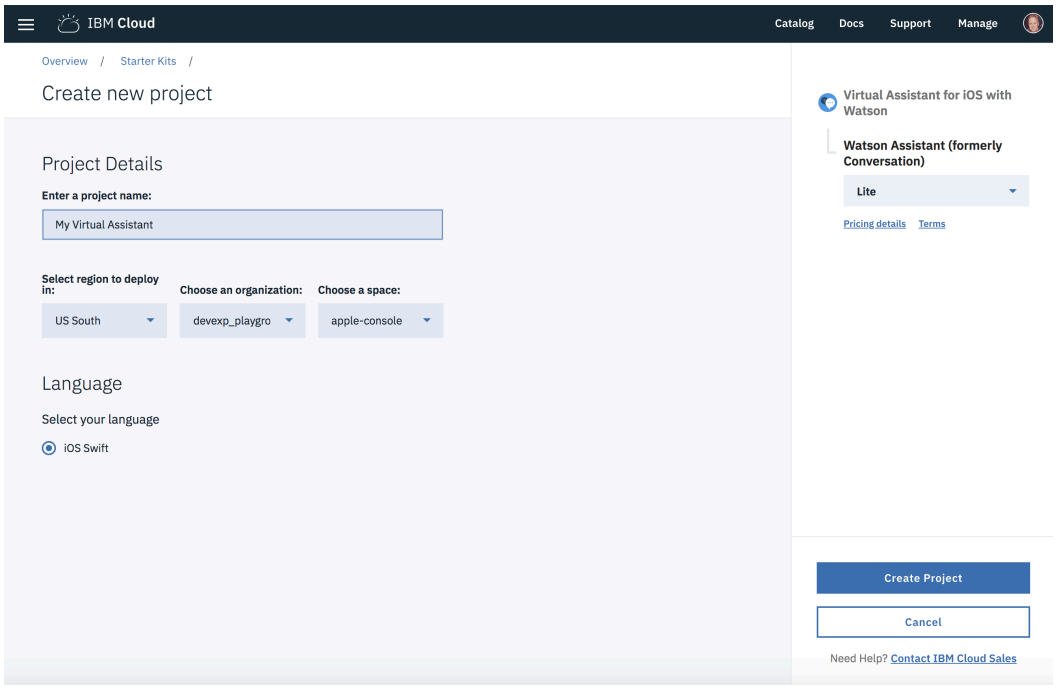
8. Search for the **Starter Kit** called **Virtual Assistant for iOS with Watson**. This starter will automatically provision an instance of the Watson Assistant service and generate a basic mobile chat application configured to use the Watson Assistant service.



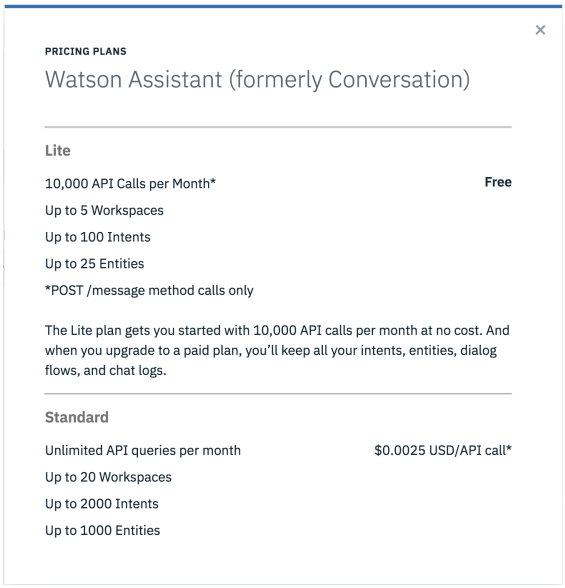
9. Each Starter Kit in the catalog of Starter Kits will have a view that describes it in detail. This is important for the first-time user to understand more detail of the Starter Kit they have chosen. Feel free to review this page to learn more about the project that will be generated. When you're ready, click on **Create app** button.



10. After clicking on the **Create app** you will be presented with the Create New Project view. You can see the IBM Cloud services that will be automatically created for you in your account. It is worth studying the **Pricing details** to understand the developer friendly **Lite** plans.

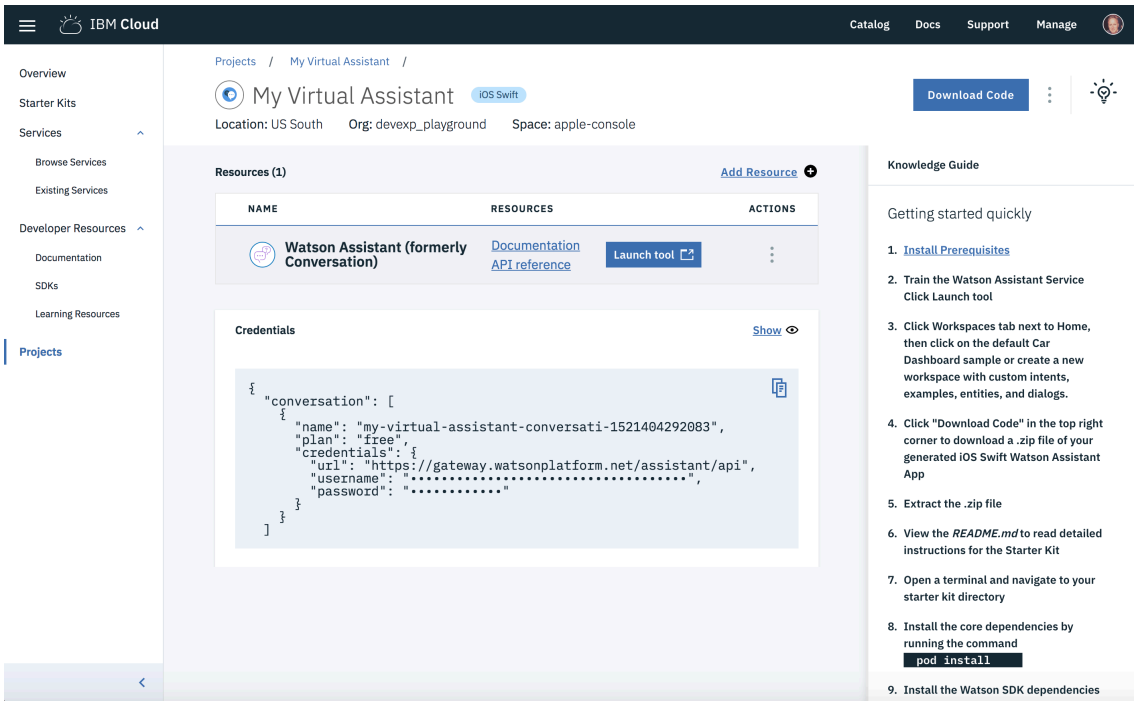


11. If you click on **Pricing details** link for the **Watson Assistant** lite plan, then you can see the service enables 10,000 API calls for free per month. These plans are optimized for the developer to get started quickly and learn without cost.

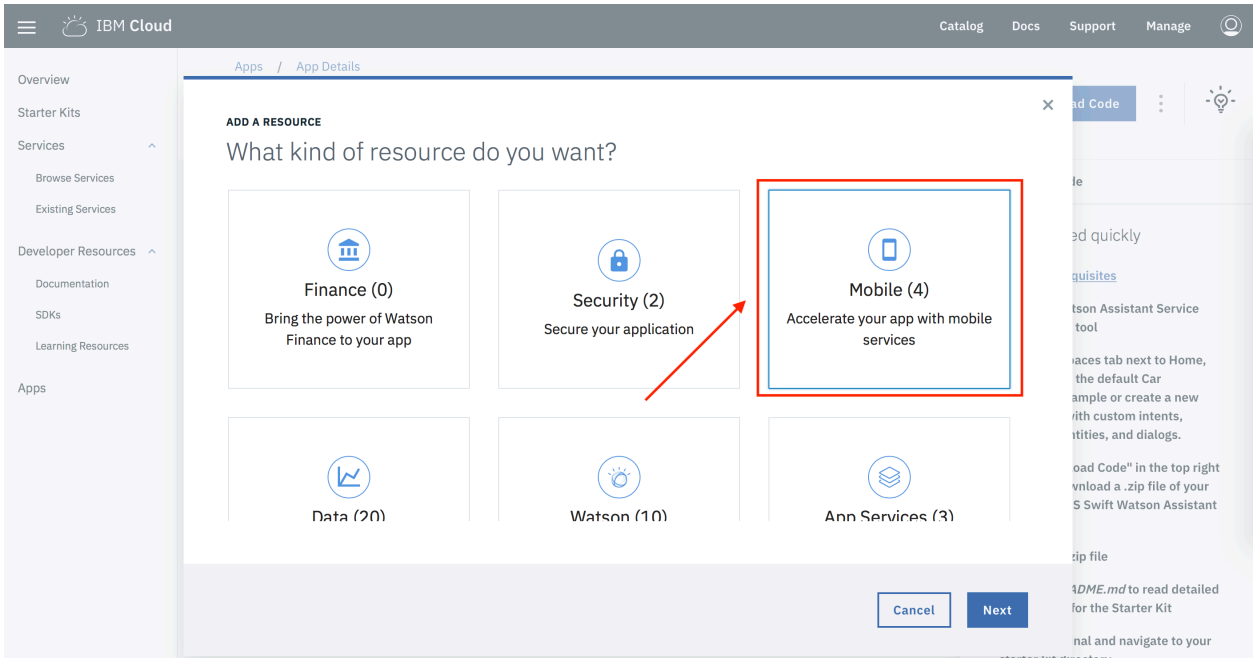


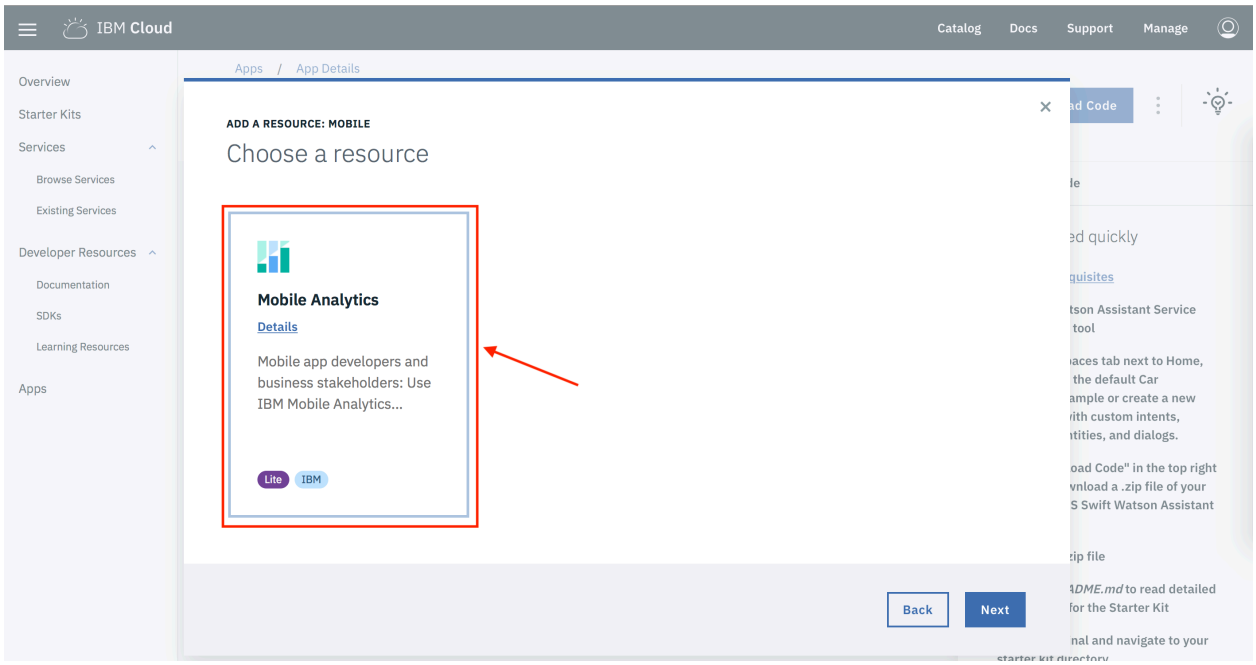
12. Enter a name **“My Virtual Assistant”** and see **iOS Swift** is already selected as the language. Click on the **Create Project** button to generate the iOS application. The App is being stored in your **Org** and **Space** that you created when you signed in.

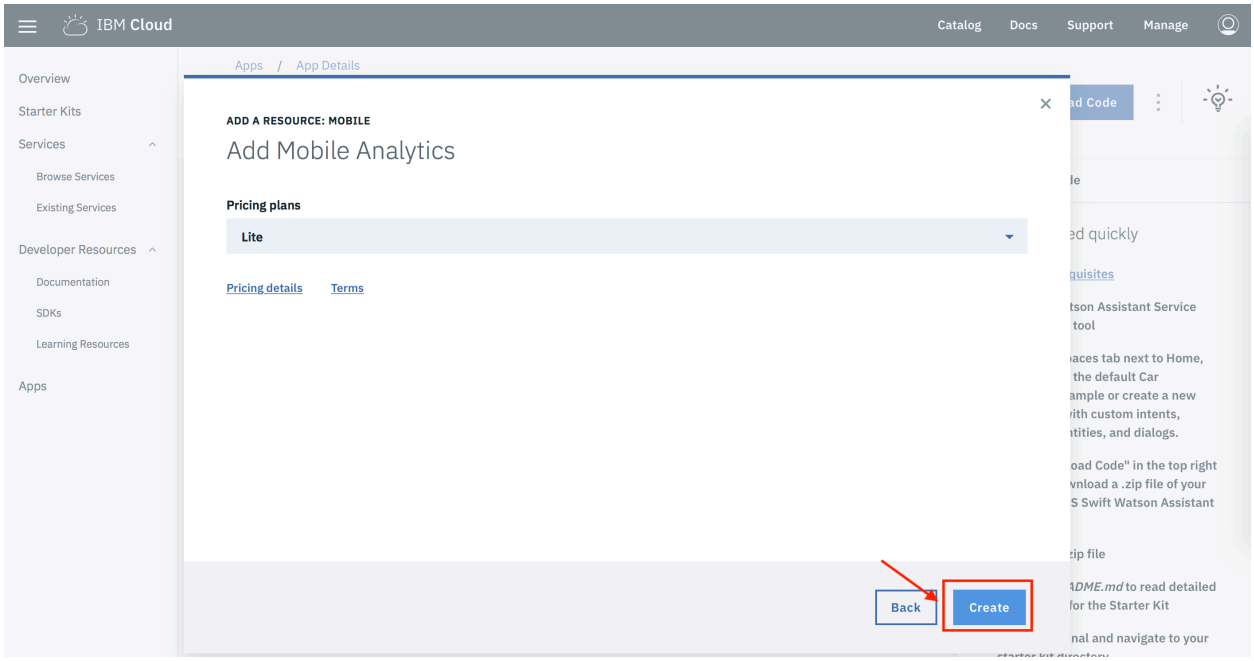
13. On the App Overview screen, you will see that the service has been automatically created for you and associated with your app. You can add other services to your app by clicking on the **Add Resource** link. You’ll see other services that support **Lite** plans. You’ll see other services with **Free** tiers that you can use if you upgrade your account by providing a credit card number. The services with **Free** tiers are designed for developers, but not intended for use in production/high scale scenarios.



14. From the resources page add the **Mobile Analytics** service to be found in the Mobile resources.

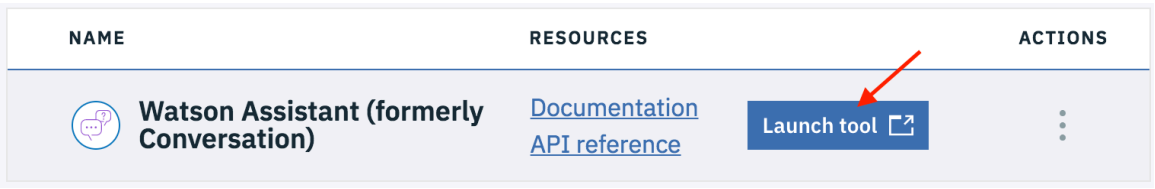

IBM Cloud
Catalog
Docs
Support
Manage
Overview
Starter Kits
Services
Browse Services
Existing Services
Developer Resources
Documentation
SDKs
Learning Resources
Apps
Apps / App Details
ADD A RESOURCE
What kind of resource do you want?
Finance (0)
Bring the power of Watson Finance to your app
Security (2)
Secure your application
Mobile (4)
Accelerate your app with mobile services
Data (20)
Watson (10)
App Services (3)
Cancel
Next


IBM Cloud
Catalog
Docs
Support
Manage
Overview
Starter Kits
Services
Browse Services
Existing Services
Developer Resources
Documentation
SDKs
Learning Resources
Apps
Apps / App Details
ADD A RESOURCE: MOBILE
Choose a resource
Mobile Analytics
Details
Mobile app developers and business stakeholders: Use IBM Mobile Analytics...
Lite
IBM
Back
Next

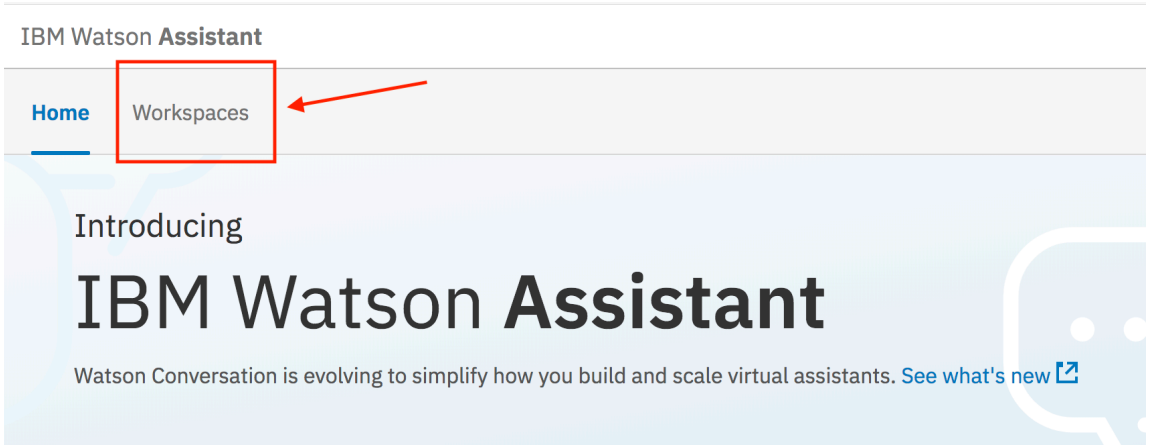


15. On the right hand side of the screen you will see a light bulb icon/button. If you click this button, you will toggle the **Knowledge Guide**. The **Knowledge Guide** provides additional information to walk you through the process of setting up your newly created project.

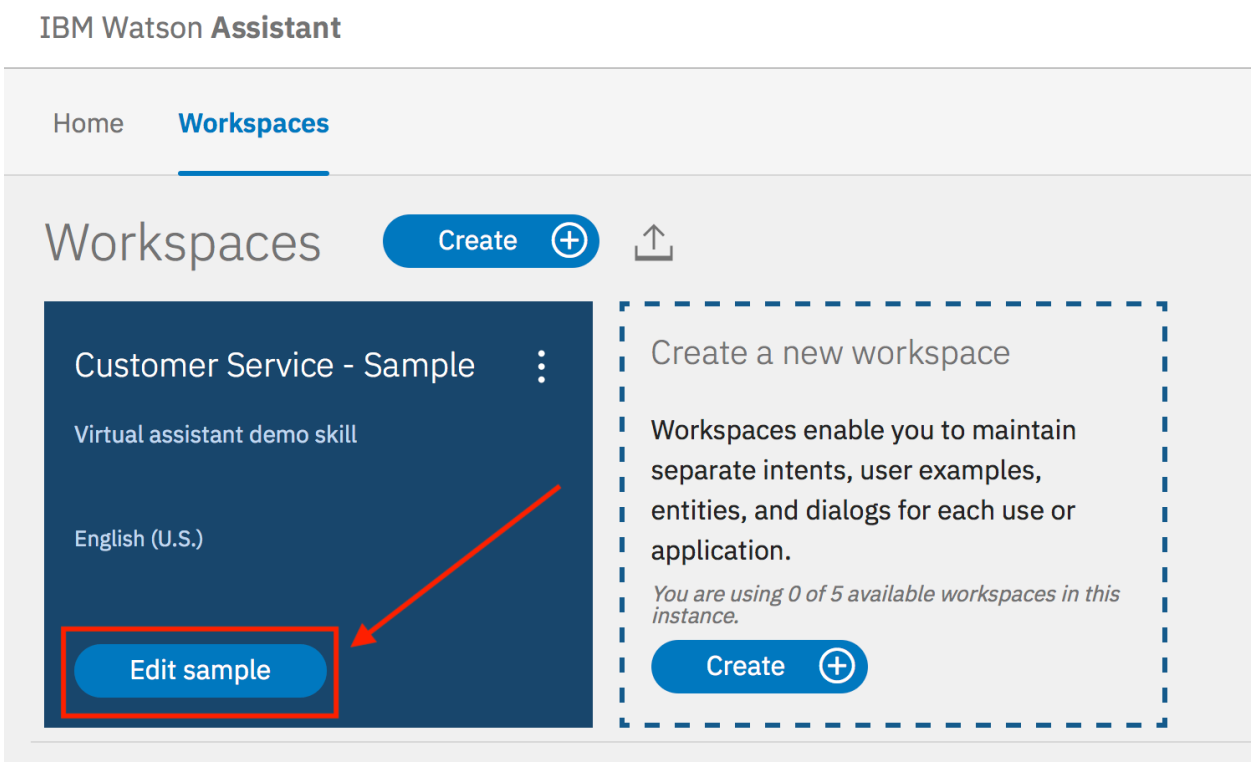
16. When the application was created, an instance of the **Watson Assistant** service was also provisioned for use. Click on the **Launch Tool** button to open the Watson Assistant workspaces tool.



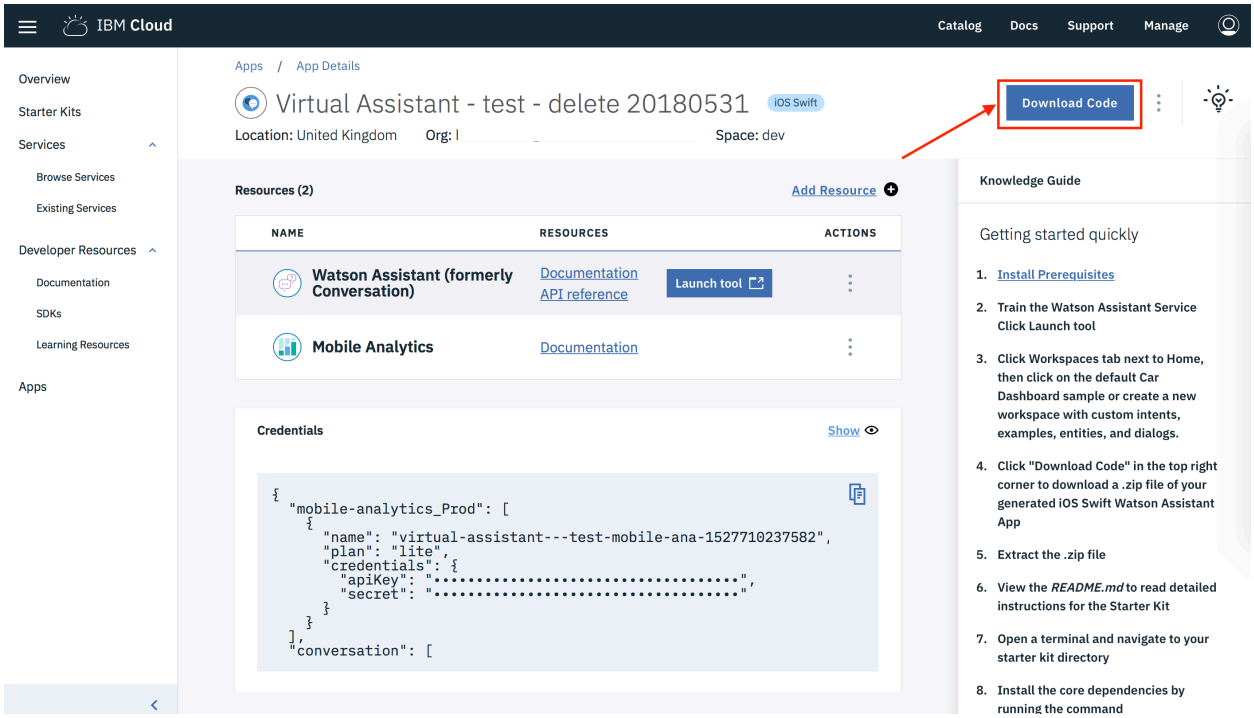
17. You need to click on the Workspaces tab. Here you can see all of your Watson Assistant workspaces.



18. The application will use the "Customer Service – Sample" workspace by default. Click on the "**Customer Service – Sample**" workspace, and you can view the intents (things you want to do), entities (things), and dialog (conversation flow) that the Watson Assistant service will be able to handle.



19. After browsing the Assistant Workspace, go back to the application overview screen and click on the **Download Code** button to download a .zip file containing the generated iOS application.



20. Extract the contents of the zip file on your local machine. Here you'll also find a README.md file that contains additional information about your application.

21. Open up a terminal and navigate to your extracted project's folder.

    The next steps assume that you already have Xcode, CocoaPods, and Carthage dependencies installed. If you don't already have these installed, you will need to install the dependencies listed earlier in ***Section 2****.*

22. Within the terminal run the "**pod install**" command to install the IBM Cloud Mobile services SDK and dependencies.

```
Last login: Sun Mar 18 09:06:58 on ttys001
13:22:18 ~/Downloads/MyVirtualAssistant-Swift $ pod install
/usr/local/lib/ruby/gems/2.3.0/gems/cocoapods-1.3.1/lib/cocoapods/executable.rb:89: warn
ing: Insecure world writable dir /usr/local/bin in PATH, mode 040777
Analyzing dependencies
Downloading dependencies
Installing BMSAnalyticsAPI (2.2.3)
Installing BMSCore (2.3.5)
Installing MessageKit (0.13.1)
Installing NVActivityIndicatorView (4.1.1)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `MyVirtualAssistant.xcworkspace` for
 this project from now on.
Sending stats
Pod installation complete! There are 3 dependencies from the Podfile and 4 total pods in
stalled.

[!] Automatically assigning platform ios with version 10.0 on target MyVirtualAssistant
because no platform was specified. Please specify a platform for this target in your Pod
file. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.
13:22:27 ~/Downloads/MyVirtualAssistant-Swift $
```

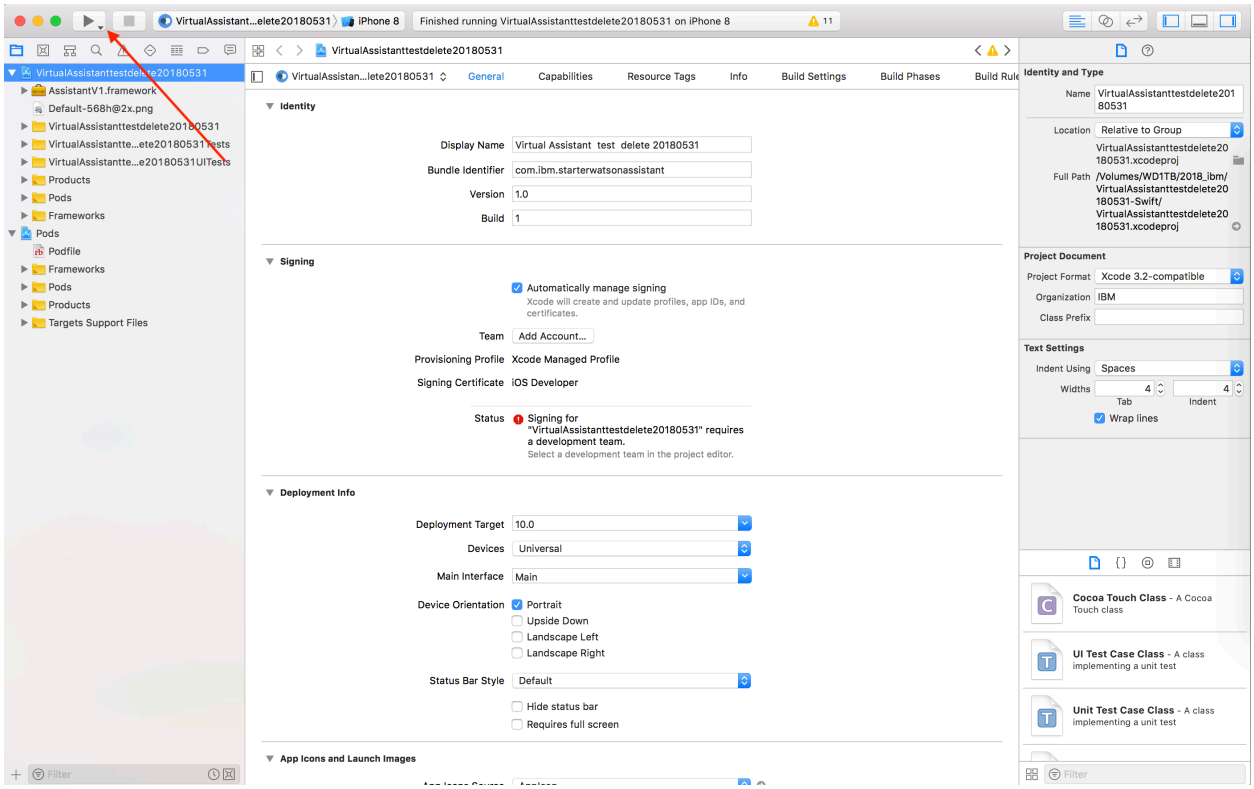
Note: If you encounter an error message "`Cocoapods: Failed to connect to GitHub to update the CocoaPods/Specs specs repo`", then you will need to update openssl and ruby on your system. You can find details how to do this here.

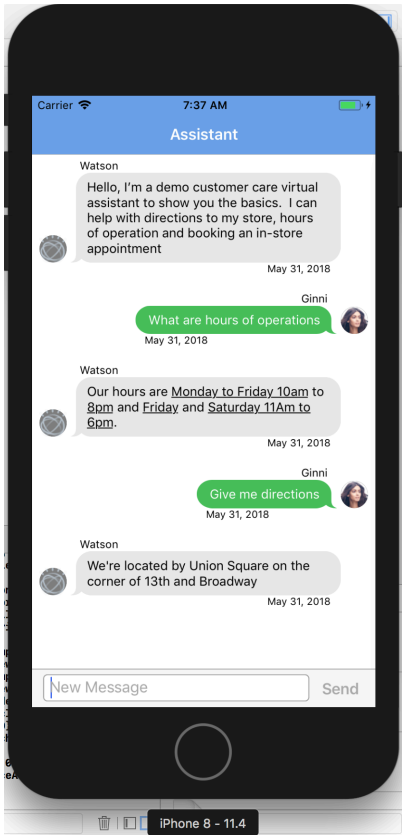
23. Once that process is complete, run the **"carthage update --platform iOS"** command to install the IBM Watson SDK and dependencies.

```
09:13:52 ~/Downloads/VirtualAssistantforiOSwithWatsonTNMXB-Swift $ carthage update --platform iOS
*** Fetching swift-sdk
*** Fetching Starscream
*** Fetching common-crypto-spm
*** Fetching zlib-spm
*** Checking out zlib-spm at "1.1.0"
*** Checking out Starscream at "3.0.4"
*** Downloading swift-sdk.framework binary at "v0.22.0"
*** Checking out common-crypto-spm at "1.1.0"
*** xcodebuild output can be found in /var/folders/c5/pmn5l8mj7g515vpzkj8ftyg40000gp/T/carthage-xcodeb
uild.JwflAU.log
*** Building scheme "Starscream" in Starscream.xcodeproj
13:25:11 ~/Downloads/VirtualAssistantforiOSwithWatsonTNMXB-Swift $
```

24. Next, open the project's `.xcworkspace` file with Xcode.

25. You're now ready to launch the app on the iOS Simulator and start interacting with the Watson Assistant service. Within Xcode, click on the **Run** button on the top bar. Once the application is running inside of the iOS Simulator, you can interact with it using natural language text. For example, you can enter "What are hours of operation", or "Give me directions".


VirtualAssistant...elete20180531
iPhone 8
Finished running VirtualAssistanttestdelete20180531 on iPhone 8
VirtualAssistanttestdelete20180531
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Identity
Display Name
Virtual Assistant test delete 20180531
Bundle Identifier
com.ibm.starterwatsonassistant
Version
1.0
Build
1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team
Add Account...
Provisioning Profile
Xcode Managed Profile
Signing Certificate
iOS Developer
Status
Signing for "VirtualAssistanttestdelete20180531" requires a development team.
Select a development team in the project editor.
Deployment Info
Deployment Target
10.0
Devices
Universal
Main Interface
Main
Device Orientation
Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style
Default
Hide status bar
Requires full screen
App Icons and Launch Images
Identity and Type
Project Document
Project Format
Xcode 3.2-compatible
Organization
IBM
Class Prefix
Text Settings
Indent Using
Spaces
Wrap lines
Cocoa Touch Class - A Cocoa Touch class
UI Test Case Class - A class implementing a unit test
Unit Test Case Class - A class implementing a unit test


Carrier
7:37 AM
Assistant
Watson
Hello, I'm a demo customer care virtual assistant to show you the basics. I can help with directions to my store, hours of operation and booking an in-store appointment
May 31, 2018
Ginni
What are hours of operations
May 31, 2018
Watson
Our hours are Monday to Friday 10am to 8pm and Friday and Saturday 11Am to 6pm.
May 31, 2018
Ginni
Give me directions
May 31, 2018
Watson
We're located by Union Square on the corner of 13th and Broadway
May 31, 2018
New Message
Send
iPhone 8 - 11.4

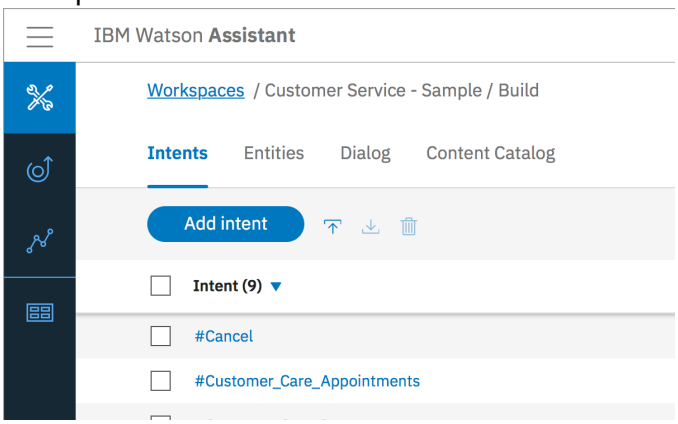
26. You're now up and running with a production-ready starter application for the Watson Assistant service! Let's update our Watson Assistant workspace to add new intents, or things that the service can recognize, to the conversation flow.

4. Modify the Watson Assistant Service Dialog

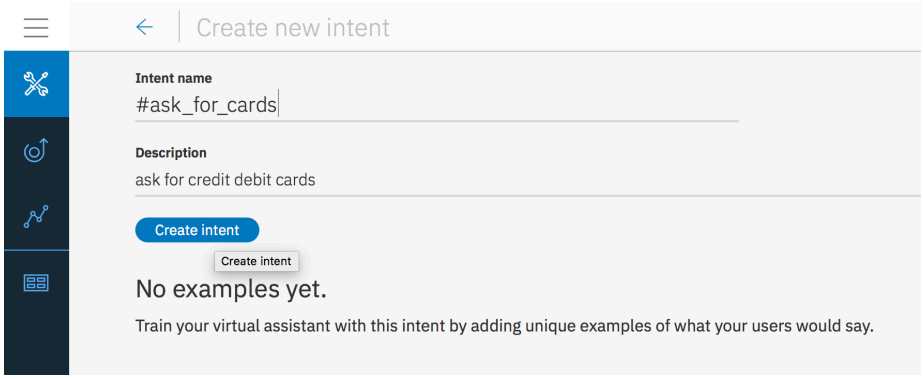
In this section, you will:

- Use the Watson Assistant workspace editor to modify the intents and structure of the **Customer Service – Sample** dialog
- You will test the conversation dialog in the web browser
- You will test the App in the iOS Simulator

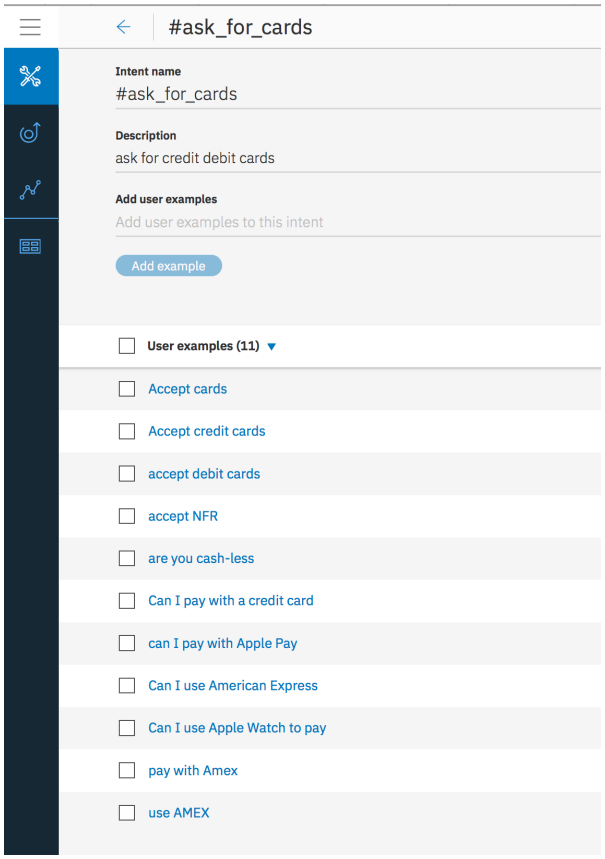
28. Go Back to the **Watson Assistant** tool and view the **Customer Service – Sample** workspace. Next, we're going to add an intent to ask for the credit/debit cards accepted. Click on the **Add intent** button.



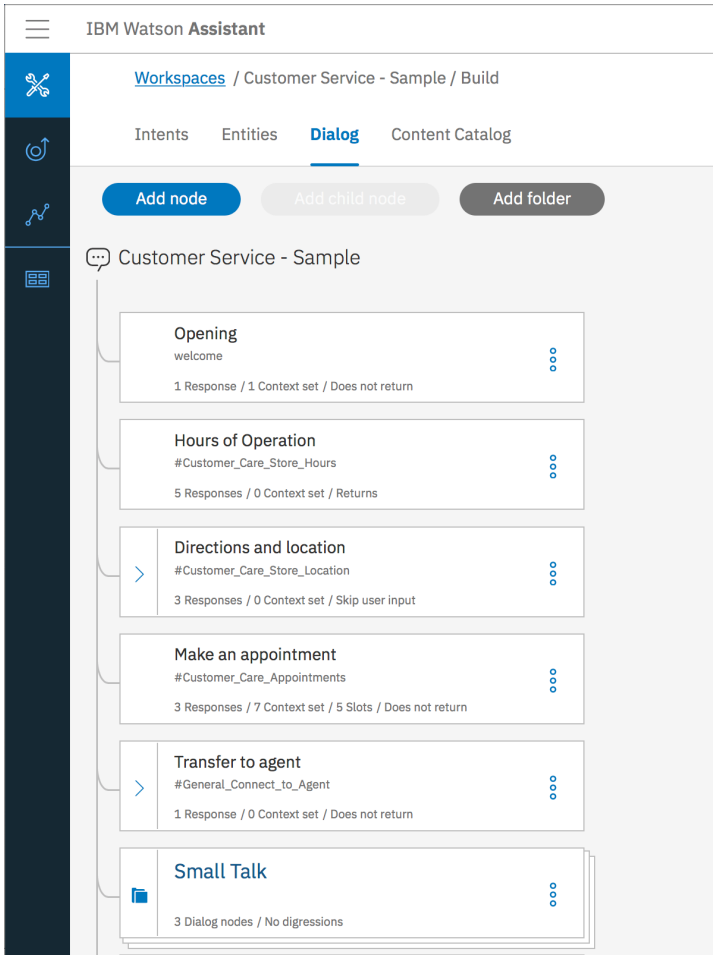
29. Now add the **intent name** "ask_for_cards" and **description** "ask for credit debit cards", then click the **Create intent** button.



30. Next, you'll need to add sample messages that the Watson Assistant service will use to train and identify the intent. Let's add these examples:
    a. "Accept credit cards"
    b. "Accept debit cards"
    c. "Accept NFR"
    d. "are you cash-less"
    e. "Can I pay with a credit card"
    f. "can I pay with Apple Pay"
    g. "Can I use American Express"
    h. "Can I use Apple Watch to pay?"
    i. "pay with Amex"
    j. "use AMEX"

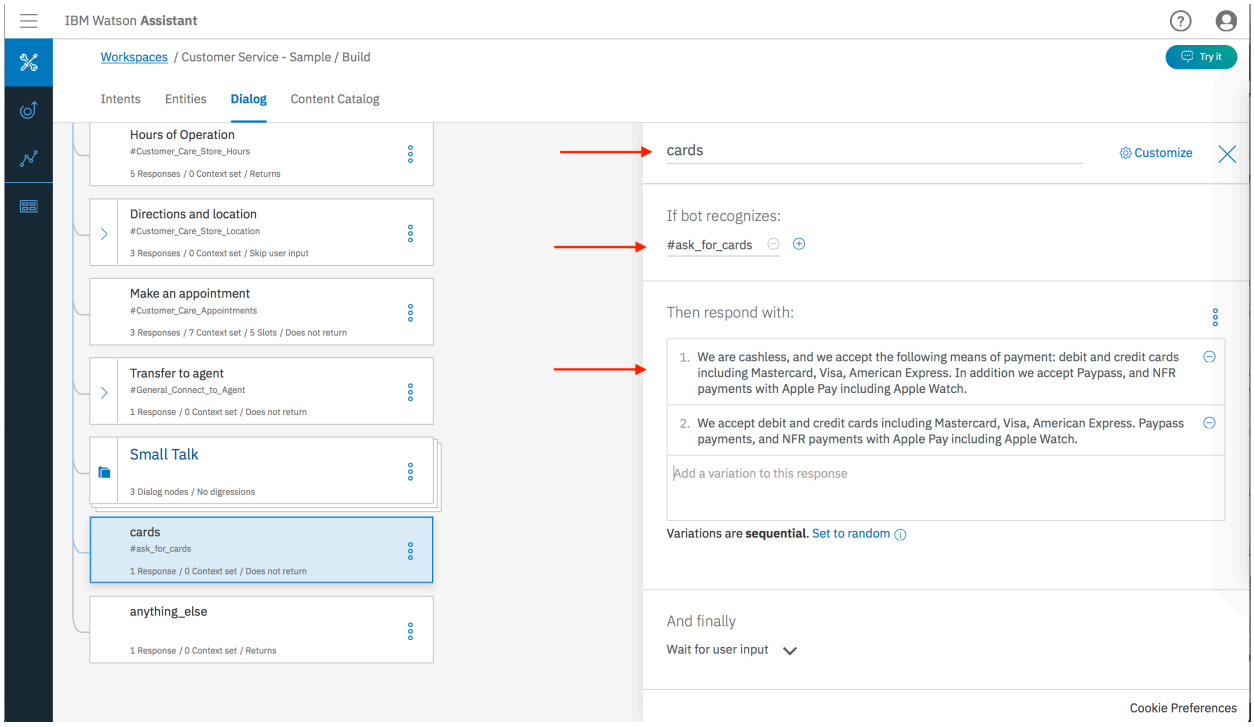


31. Go back to the **Customer Service – Sample** workspace and click on the **Dialog** link/tab. Here you'll see the logical structure and flow of the conversation.

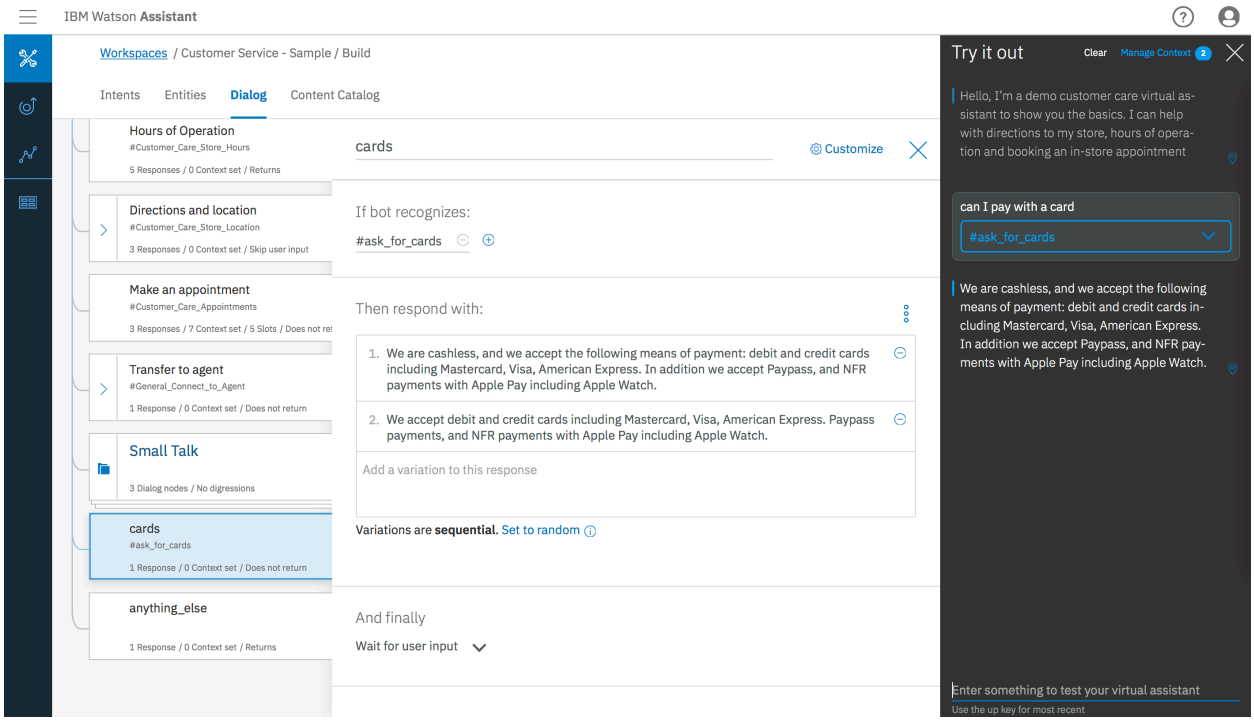


32. Click **Add node** to add a new node to the conversation flow.

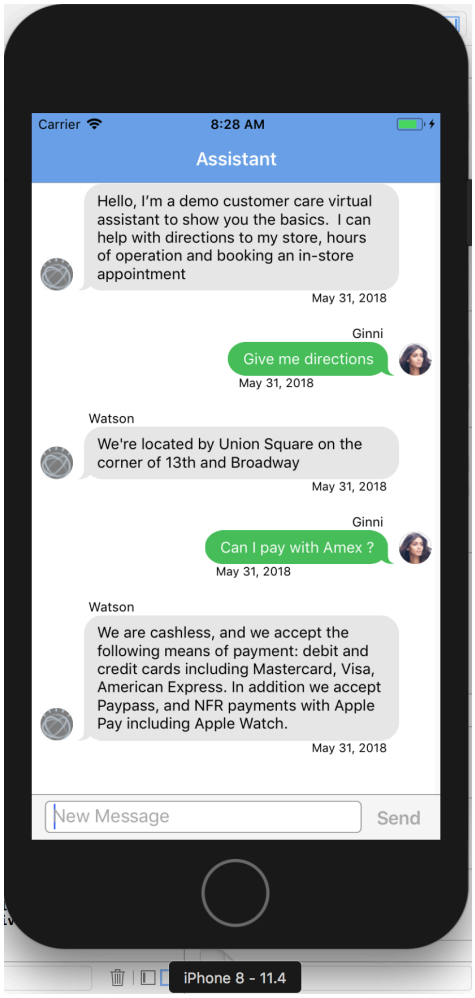
33. Enter the **name** “cards”, then specify the “#ask_for_cards” intent under **if bot recognizes**, then add responses that the service can respond with (“We accept debit and credit cards including Mastercard, Visa, American Express. Paypass payments, and NFR payments with Apple Pay including Apple Watch.”, “We are cashless, and we accept the following means of payment: debit and credit cards including Mastercard, Visa, American Express. In addition we accept Paypass, and NFR payments with Apple Pay including Apple Watch.”). These responses could also include parameters or system values that are part of the conversation flow.



34. Now, let's test out the new conversation flow in the browser. Click **Try it** in the top right corner of the Watson Assistant workspace. Here you'll have a sample chat dialog where you can enter freeform text to test the flow of the conversation dialog. Enter the value "Can I pay with a card" and see what gets returned.



35. We can see that our conversation dialog has been updated. Next, we'll test that in the iOS Simulator. Go back to the **iOS Simulator** and submit the text "Can I pay with Amex" and verify the results. Feel free to make variations on the text to experiment with how the Watson Assistant service will respond.



5. Mobile Analytics

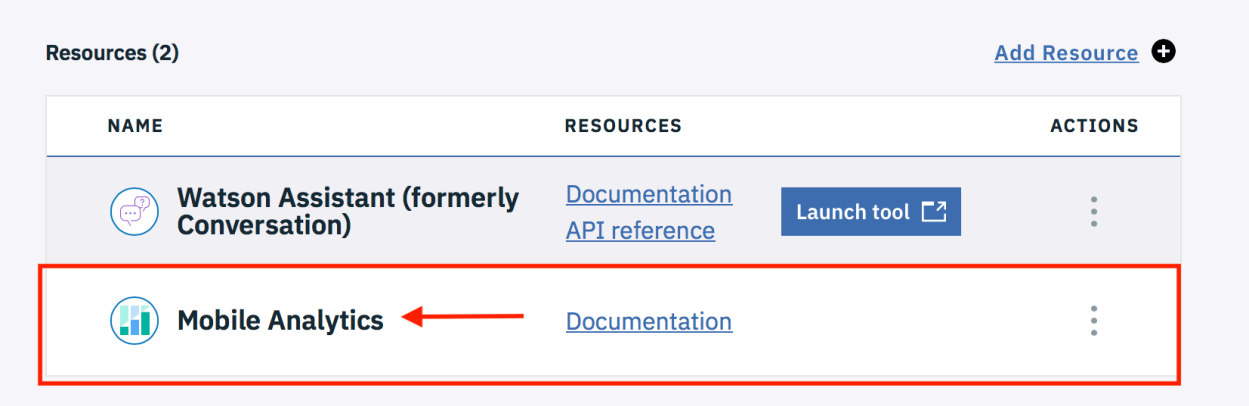
The IBM Mobile Analytics for IBM Cloud provides monitoring and analytics for your mobile applications. You can record application logs and monitor data with the Mobile Analytics Client SDK. Developers can control when to send this data to the Mobile Analytics Service. When data is delivered to Mobile Analytics, you can use the Mobile Analytics console to get analytics insights about your mobile applications, devices, and application logs.

Check for more information here:
https://console.bluemix.net/docs/services/mobileanalytics/app-monitoring-metrics.html

Change the level of logging to analytics in class AppDelegate (file AppDelegate.swift) in func Application:

```
Logger.logLevelFilter = .debug
```

You can observe the results of application starts and eventual crashes at the monitoring from the Mobile Analytics dashboard



6. Conclusion

Congratulations! You have completed the lab.

You have successfully used the IBM Cloud to create a simple Watson Assistant app for iOS using the Swift language and the Watson Assistant service.

Contacts

Emma Tucker (etucker@us.ibm.com)
Offering Management, IBM Cloud Developer Services

Matthew Perrins (mjperrin@us.ibm.com)
STSM, Architect IBM Cloud Developer Services

Andrew Trice (amtrice@us.ibm.com)
Technical Product Manager/Architect, IBM Cloud Developer Services

Marek Sadowski (mwsadows@us.ibm.com)
Developer Advocate, IBM DBG

Acknowledgements and Disclaimers