

IBM Cloud

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

| | |
|---|-----------|
| CREATE VIRTUAL ASSISTANT FOR IOS WITH THE IBM CLOUD APPLE DEVELOPMENT CONSOLE AND WATSON | 1 |
| 1. INTRODUCTION | 3 |
| 2. SETUP | 4 |
| 3. CREATE AN IOS SWIFT APP AND USE THE WATSON ASSISTANT SERVICE | 8 |
| 4. MODIFY THE WATSON ASSISTANT SERVICE DIALOG | 22 |
| 5. CONCLUSION | 27 |
| Contacts | 28 |
| ACKNOWLEDGEMENTS AND DISCLAIMERS | 29 |



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.



Sign up for an IBMid and create your Bluemix account
Build on Bluemix for free with no time restrictions

Guaranteed free development with Lite plans
Develop worry-free and at no cost with cap based Lite plan services for as long as you like.

Start on your projects right away
Skip entering your credit card info and get working in just a few short steps.

Get \$200 on us to try paid services
Ease into cloud pricing or try something new with \$200 in credit available for 1 month upon upgrade.

Ready to get started? Sign up today!

Email*
mattjperrin@gmail.com

First Name*
Matthew

Last Name*
Perrins

Company
IBM Lab

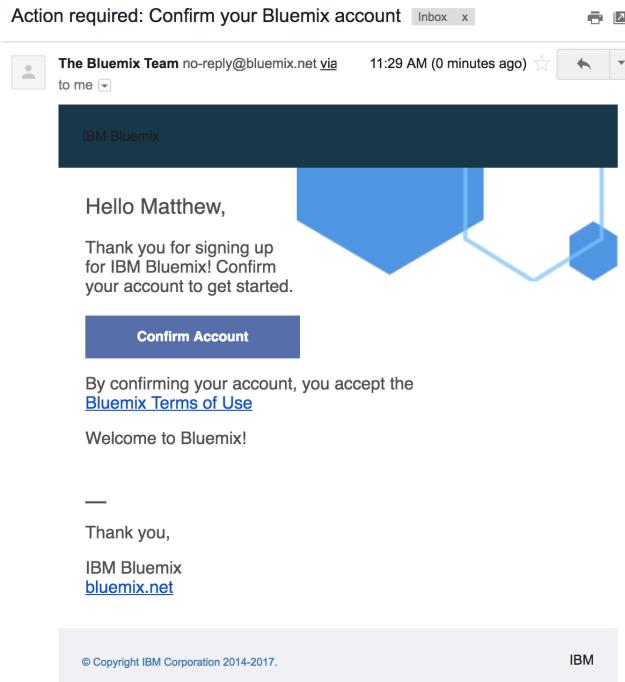
Country or Region*
United States

Phone Number*

Password*
[Forgot Password?](#)

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.

The screenshot shows the IBM Cloud Dashboard. At the top, there's a navigation bar with 'IBM Cloud' and links for Catalog, Docs, Support, Manage, and a user profile icon. Below the navigation, the dashboard has several sections:

- Watson**: Includes Conversation, Discovery, Language Translator, Natural Language Understanding, Personality Insights, and Tone Analyzer.
- Internet of Things**: Includes Internet of Things Platform.
- Mobile**: Includes Push Notifications.
- DevOps**: Includes Availability Monitoring.
- Apps**: Includes Liberty for Java™, SDK for Node.js™, ASP.NET Core, Runtime for Swift, XPages, Go, PHP, Python, Ruby, and Tomcat.

In the bottom right corner, there's a message from 'Corrie from IBM Cloud': "Hey Matthew, How important are Continuous Integration and Continuous Delivery to your...". A blue circular icon with a white speech bubble and a red notification badge (with the number 2) is also visible.

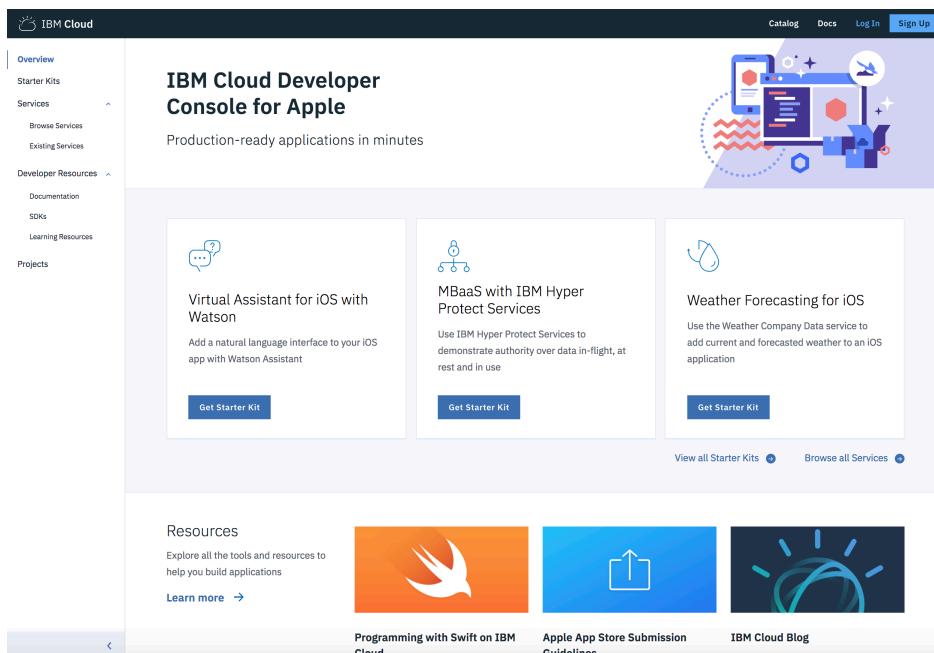


3. Create an iOS Swift App and use the Watson Assistant Service

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.

The screenshot shows a grid of three cards, each representing a different Starter Kit:

- Virtual Assistant for iOS with Watson**: Features a speech bubble icon. Description: "Add a natural language interface to your iOS app with Watson Assistant". Call-to-action: "Get Starter Kit".
- MBaaS with IBM Hyper Protect Services**: Features a lock icon. Description: "Use IBM Hyper Protect Services to demonstrate authority over data in-flight, at rest and in use". Call-to-action: "Get Starter Kit".
- Weather Forecasting for iOS**: Features a weather icon. Description: "Use the Weather Company Data service to add current and forecasted weather to an iOS application". Call-to-action: "Get Starter Kit".

At the bottom of the screen, there are two navigation links: "View all Starter Kits" and "Browse all Services".

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.

The Resources section includes the following links:

- Resources**: "Explore all the tools and resources to help you build applications". Call-to-action: "Learn more →".
- Programming with Swift in IBM Cloud**: "This programming guide will help you understand how to build applications for the iOS, Swift, and IBM Cloud environments". Call-to-action: "More".
- Apple App Store Submission Guidelines**: "Provide a great app experience and effectively communicate your App Store affiliation by following these guidelines and best practices". Call-to-action: "More".
- IBM Cloud Blog**: "Keep up to date with the latest from IBM Cloud on our blog". Call-to-action: "More".

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.



Join the IBM Apple Community

Get answers to your questions, connect with experts and be part of the community.



dW Answers

Connect with the Apple developer community and IBM Swift experts for all of your technical questions



Slack

Join the community of IBM Apple developers and learn from your peers how to get the most out of IBM Cloud APIs and SDKs.



Stack Overflow

Connect with the IBM developer community for all your 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.

The screenshot shows the IBM Cloud Apple Development Console interface. The left sidebar is collapsed, and the main content area is titled "Software Development Kits". It lists several services with their respective SDKs:

- Analytics**: iOS Swift. Description: Analytics SDK provides insight into how your apps are performing and how they are being used. Links: View source, Programming Guide.
- Notifications**: iOS Swift. Description: Client SDK Swift Push. Links: View source, Programming Guide.
- Engagement**: Swift. Description: The App Launch service helps in controlled reach of app features, customize your apps. Links: View source.
- Watson**: iOS Swift. Description: The Watson Swift SDK enables developers to quickly add Watson Cognitive Computing services to their Swift. Links: View source, Programming Guide.
- Authentication**: iOS Swift. Description: IBM Cloud App ID iOS Swift SDK. Links: View source, Programming Guide.
- Database - NoSQL**: Swift. Description: Cloud Swift SDK use swift-cloudant to store, index and query remote JSON data on Cloudant, CouchDb. Links: View source.

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.

The screenshot shows the IBM Cloud Apple Development Console interface. On the left, there is a navigation sidebar with the following items: Overview, Starter Kits (which is highlighted with a red arrow), Services, Developer Resources, and Projects. The main content area is titled "Select an iOS or Server-Side Swift Starter Kit" and contains the following options:

- Create Project**: Start creating a custom application using services of your choice. This option is not highlighted.
- Virtual Assistant for iOS with Watson**: Use a Watson Assistant service to add a natural language interface to your iOS application. This option is highlighted with a red box.
- Basic MBaaS for iOS**: Create an iOS Backend as a Service with Push, Analytics and NoSQL data. This option is partially visible on the right.
- Swift for Backend for Frontend API**: Begin building backend-for-frontend APIs in Swift, using the Kitura framework and a Cloudant NoSQL... This option is partially visible at the bottom left.
- Tone Analyzer for iOS with Watson**: Use Watson's Tone Analyzer deep learning capabilities to evaluate your passages of text. Type in an... This option is partially visible at the bottom center.
- Infinite Scrolling with Cloudant NoSQL for iOS**: Visualize your Cloudant database as an infinitely scrolling list. This option is partially visible at the bottom right.

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.



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

The screenshot shows the IBM Cloud Apple Development Console interface. On the left, a sidebar menu includes 'Overview', 'Starter Kits' (which is selected and highlighted in blue), 'Services', 'Developer Resources', and 'Projects'. The main content area is titled 'Select an iOS or Server-Side Swift Starter Kit' and contains several cards for different starter kits:

- Create Project**: A card with a large plus sign icon, describing it as a way to start creating a custom application using services of choice.
- Virtual Assistant for iOS with Watson**: Describes using a Watson Assistant service to add a natural language interface to an iOS application. It includes 'Lite' and 'Mobile App' options.
- Basic MBaaS for iOS**: Describes creating an iOS Backend as a Service with Push, Analytics and NoSQL data. It includes 'Lite' and 'Mobile App' options.
- Swift for Backend for Frontend API**: Describes begin building backend-for-frontend APIs in Swift, using the Kitura framework and a Cloudant NoSQL database. It includes 'Lite' and 'Backend for Frontend' options.
- Tone Analyzer for iOS with Watson**: Describes using Watson's Tone Analyzer deep learning capabilities to evaluate your passages of text. It includes 'Lite' and 'Mobile App' options.
- Infinite Scrolling with Cloudant NoSQL for iOS**: Describes visualizing your Cloudant database as an infinitely scrolling list. It includes 'Lite' and 'Mobile App' options.

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.

The screenshot shows the 'Starter Kit Details' page for the 'Virtual Assistant for iOS with Watson' kit. The top navigation bar includes 'Catalog', 'Docs', 'Support', 'Manage', and a user profile icon. The sidebar on the left is identical to the one in the previous screenshot. The main content area has the following sections:

- Title**: Virtual Assistant for iOS with Watson
- Type**: Mobile App • Lite
- Overview**: A brief description of how to quickly build a virtual assistant across various channels using the Apple platform developer tools.
- Services used**: Watson Assistant (formerly Conversation)
- This starter kit will help you**: A bulleted list of benefits:
 - Get started quickly and easily. Get faster time to value, and integrate across channels, networks and environments.
 - Utilize Watson Assistant's reliable infrastructure that scales with individual use cases. Platform support from IBM gives you the backing you need.
 - Own your data. IBM protects your privacy, allowing you to opt out of data sharing. Built on IBM Cloud and featuring reliable tooling with industry-leading security.

A red arrow points to the 'Create app' button in the top right corner of the main content area.

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.

The screenshot shows the 'Create new project' screen in the IBM Cloud interface. On the left, under 'Project Details', a project name 'My Virtual Assistant' is entered. Deployment settings are set to 'US South', 'devexp_playgro', and 'apple-console'. Under 'Language', 'iOS Swift' is selected. On the right, a sidebar lists services: 'Virtual Assistant for iOS with Watson' and 'Watson Assistant (formerly Conversation)'. The 'Watson Assistant' section is expanded, showing the 'Lite' plan selected. A link to 'Pricing details' is visible. At the bottom right are 'Create Project' and 'Cancel' buttons.

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.

The dialog box shows the 'PRICING PLANS' for 'Watson Assistant (formerly Conversation)'. It lists two plans: 'Lite' and 'Standard'. The 'Lite' plan is free and includes 10,000 API Calls per Month*, Up to 5 Workspaces, Up to 100 Intents, and Up to 25 Entities. It also notes that *POST /message method calls only. The 'Standard' plan costs \$0.0025 USD/API call* and includes Unlimited API queries per month, Up to 20 Workspaces, Up to 2000 Intents, and Up to 1000 Entities. A note at the bottom states: 'The Lite plan gets you started with 10,000 API calls per month at no cost. And when you upgrade to a paid plan, you'll keep all your intents, entities, dialog flows, and chat logs.'



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.

Knowledge Guide

1. [Install Prerequisites](#)
2. Train the Watson Assistant Service
Click Launch tool
3. Click Workspaces tab next to Home, then click on the default Car Dashboard sample or create a new workspace with custom intents, examples, entities, and dialogs.
4. Click "Download Code" in the top right corner to download a .zip file of your generated iOS Swift Watson Assistant App
5. Extract the .zip file
6. View the *README.md* to read detailed instructions for the Starter Kit
7. Open a terminal and navigate to your starter kit directory
8. Install the core dependencies by running the command
`pod install`
9. Install the Watson SDK dependencies

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

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**

ADD A RESOURCE: MOBILE

Choose a resource

- Mobile Analytics**

Details

Mobile app developers and business stakeholders: Use IBM Mobile Analytics...

Lite **IBM**

Back **Next**



The screenshot shows the 'IBM Cloud' interface with the 'App Details' section for 'Add Mobile Analytics'. On the left, there's a sidebar with categories like Overview, Starter Kits, Services, Developer Resources, and Apps. The main area shows a 'Pricing plans' dropdown set to 'Lite'. At the bottom right, there are two buttons: 'Back' and 'Create'. The 'Create' button is highlighted with a red box and an arrow pointing to it.

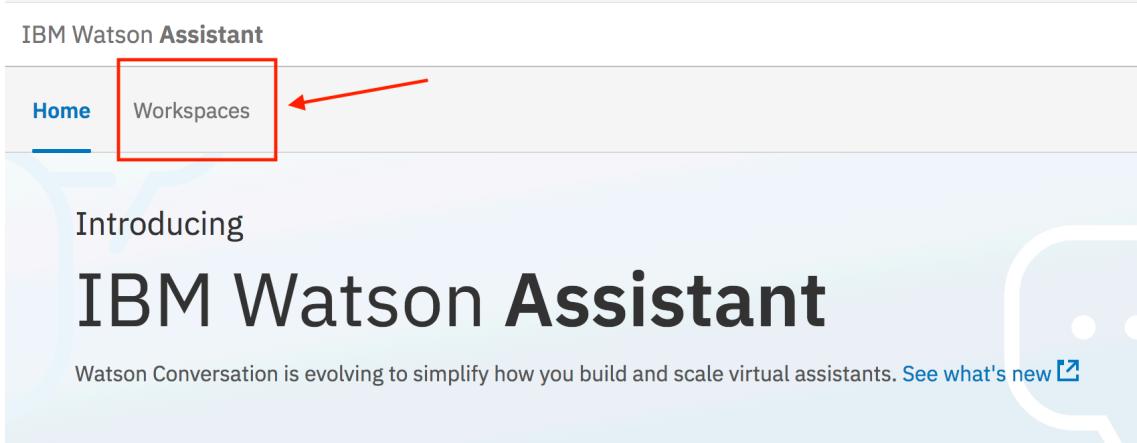
15. On the right hand side of the screen you will see a light bulb icon/button. If you click this button, you will toggle the **Knowledge Guide**. The **Knowledge Guide** provides additional information to walk you through the process of setting up your newly created project.
16. When the application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

The screenshot shows a table listing the 'Watson Assistant (formerly Conversation)' service. The table has columns for NAME, RESOURCES, and ACTIONS. Under 'RESOURCES', there are links to 'Documentation' and 'API reference'. Under 'ACTIONS', there is a blue button labeled 'Launch tool' with a red arrow pointing to it. There are also three vertical dots in the 'ACTIONS' column.

| NAME | RESOURCES | ACTIONS |
|--|--|---------------------------------|
|  Watson Assistant (formerly Conversation) | Documentation API reference | Launch 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.

A screenshot of the 'Workspaces' section of the IBM Watson Assistant interface. The top navigation bar shows 'Home' and 'Workspaces', with 'Workspaces' being the active tab. Below the navigation is a 'Create' button with a plus sign. The main area is titled 'Workspaces' and contains a card for the 'Customer Service - Sample' workspace. The card details include: 'Virtual assistant demo skill', 'English (U.S.)', and an 'Edit sample' button, which is highlighted with a red box and has a red arrow pointing to it from the bottom left. To the right of the workspace card is a dashed-line box containing descriptive text about workspaces and a second 'Create' button.

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.



The screenshot shows the IBM Cloud Apple Development Console interface. On the left, there's a sidebar with navigation links like Overview, Starter Kits, Services, Developer Resources, Documentation, SDKs, Learning Resources, and Apps. The main area displays the 'Virtual Assistant - test - delete 20180531' app details. It includes sections for Resources (2), Credentials (JSON snippet shown), and Knowledge Guide. The Knowledge Guide lists steps for creating a Watson Assistant Service, Click Workspaces tab, Download Code, Extract the .zip file, View README.md, Open a terminal, and Install dependencies.

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: warning: 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 installed.

[!] 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 Podfile. 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](#).

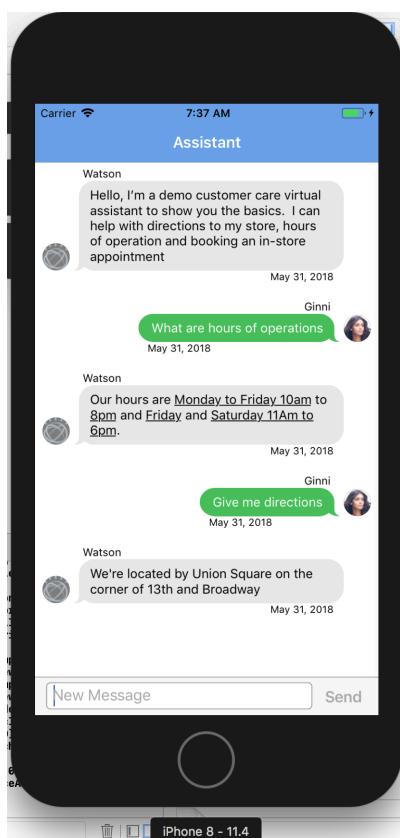
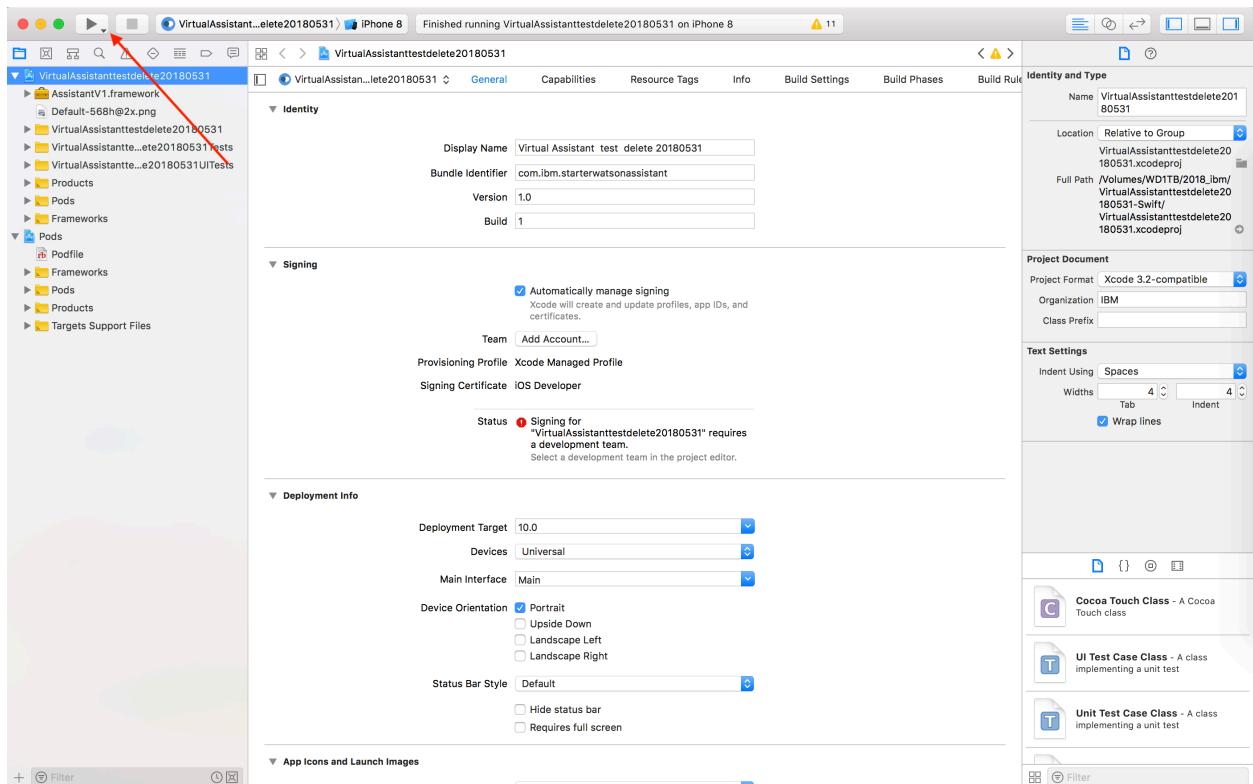
23. Once that process is complete, run the “**carthage update --platform iOS**” command to install the IBM Watson SDK and dependencies.

```
09:13:52 ~/Downloads/VirtualAssistantforiOSwithWatsonTNMB-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-xcodebu
uild.JwfLAU.log
*** Building scheme "Starscream" in Starscream.xcodeproj
13:25:11 ~/Downloads/VirtualAssistantforiOSwithWatsonTNMB-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”.





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.

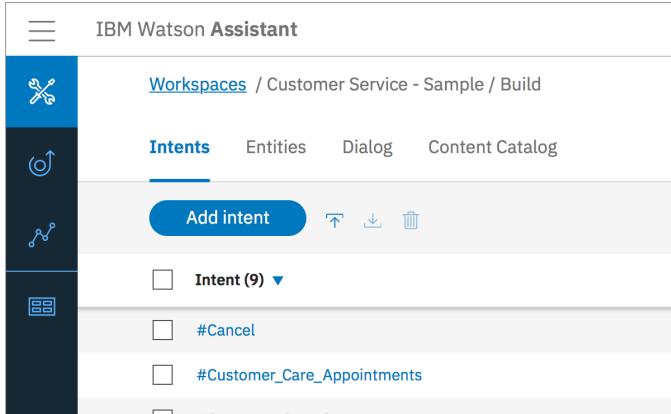


4. Modify the Watson Assistant Service Dialog

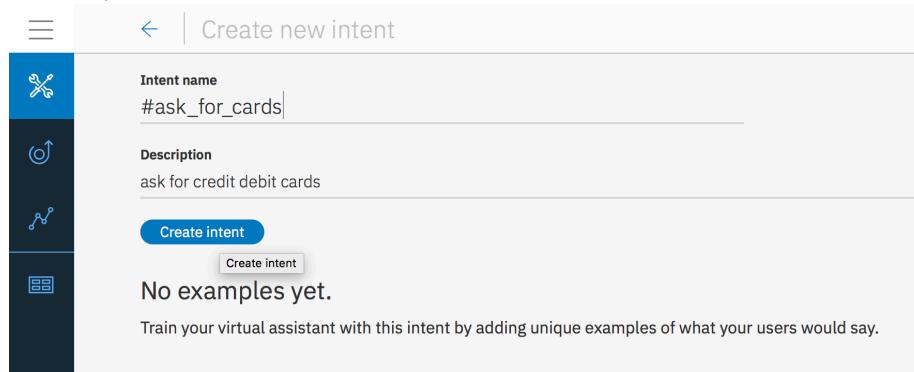
In this section, you will:

- 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:

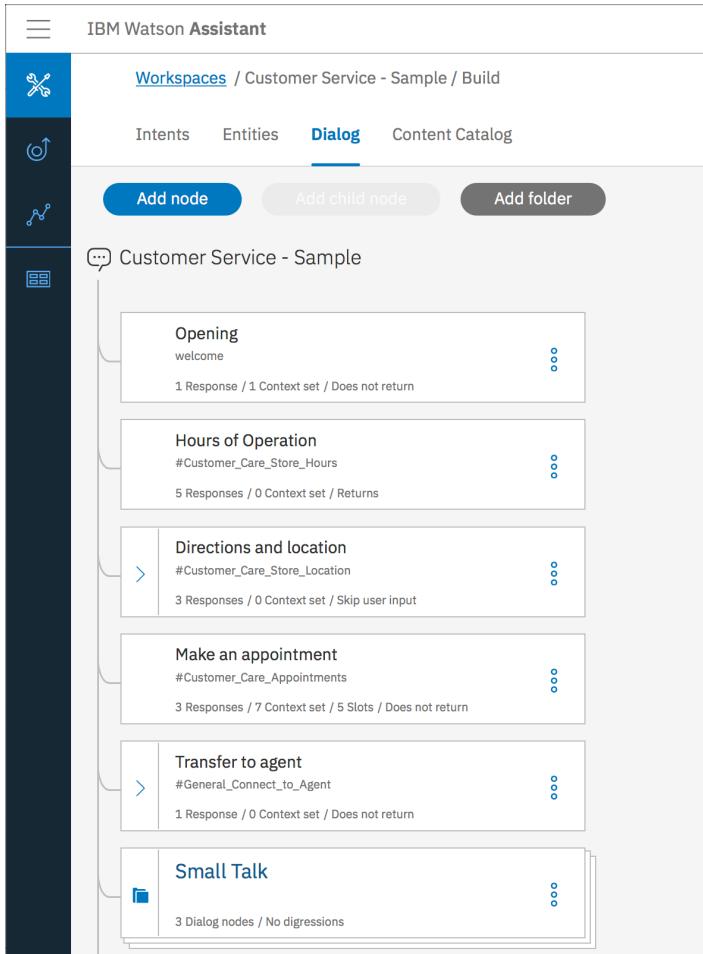
- a. "Accept credit cards"
- b. "Accept debit cards"
- c. "Accept NFR"
- d. "are you cash-less"
- e. "Can I pay with a credit card"
- f. "can I pay with Apple Pay"
- g. "Can I use American Express"
- h. "Can I use Apple Watch to pay?"
- i. "pay with Amex"
- j. "use AMEX"

The screenshot shows the configuration page for the intent '#ask_for_cards'. The left sidebar has icons for Home, Create, Manage, and Help. The main area shows the intent name '#ask_for_cards' and a description 'ask for credit debit cards'. Below that is a section for 'Add user examples' with a button 'Add example'. A list of 11 user examples is shown, each with a checkbox:

- Accept cards
- Accept credit cards
- accept debit cards
- accept NFR
- are you cash-less
- Can I pay with a credit card
- can I pay with Apple Pay
- Can I use American Express
- Can I use Apple Watch to pay
- pay with Amex
- 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.



The screenshot shows the IBM Watson Assistant workspace. On the left, there's a sidebar with icons for workspace management. The main area has tabs for 'Intents', 'Entities', 'Dialog' (which is selected), and 'Content Catalog'. In the 'Dialog' tab, a list of nodes is shown:

- 'Hours of Operation' (intent: #Customer_Care_Store_Hours)
- 'Directions and location' (intent: #Customer_Care_Store_Location)
- 'Make an appointment' (intent: #Customer_Care_Appointments)
- 'Transfer to agent' (intent: #General_Connect_to_Agent)
- 'Small Talk'
- A card node labeled 'cards' (intent: #ask_for_cards)
- 'anything_else'

Red arrows point from the 'cards' node to the 'If bot recognizes:' section and the 'Then respond with:' section. The 'If bot recognizes:' section contains the intent '#ask_for_cards'. The 'Then respond with:' section contains two variations of a response message about payment methods.

Variations are sequential. Set to random

At the bottom right, there are 'Cookie Preferences' and a 'Try it' button.

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.

The screenshot shows the 'Try it out' interface integrated into the workspace. The workspace on the left is identical to the previous screenshot. On the right, the 'Try it out' interface is open, showing a conversation window:

User: can I pay with a card

Bot: #ask_for_cards

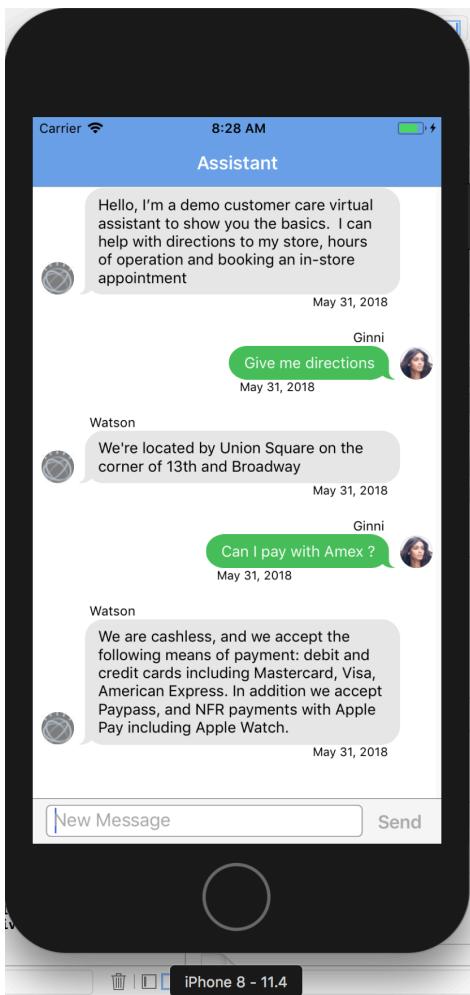
Bot response:

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

Below the conversation window, there's a text input field with placeholder text: "Enter something to test your virtual assistant" and "Use the up key for most recent".



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

Resources (2)

Add Resource +

| NAME | RESOURCES | ACTIONS |
|--|--|-------------------------------|
| Watson Assistant (formerly Conversation) | Documentation API reference | Launch tool ↗ |
| Mobile Analytics | Documentation | ⋮ |



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

Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights – use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts.

In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli® Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

