Microsoft Azure



提供元:マイクロソフト

■ 目次

Microsoft Azure の利用準備

- Microsoft Azure とは
- DreamSpark とは
- Microsoft Azure を利用するために必要な知識
- Microsoft Azure for DreamSpark の利用開始手順

サンプルアプリの紹介

- アンケートアプリ
- 匿名 SNS

Microsoft Azure の利用準備

Microsoft Azure とは

Microsoft Azure は、Web サイトを公開するための「Web サーバー」や、大量のデータを蓄積・選別し、必要 に応じて引き出す技術である「SQL データベース」などクラウド上で利用できるさまざまな機能を包括した サービスの総称です。モバイルアプリで利用するデータを管理する機能は「App Service」という名称で提供 されています。ここでは Monaca で作ったアプリのデータを App Service に登録するサンプルアプリを紹介し ます。

DreamSpark とは

DreamSpark は、マイクロソフトの学校向け教育支援プログラムです。Office や Visual Studio などの製品が 学生向けに無償で提供されています。今回サンプルアプリで利用する Microsoft Azure も、DreamSpark の製 品ラインナップの中に含まれています。なお、対象となるのは高等学校、専門学校、専修学校、高等専門学校、 大学に所属している学生です。学習支援を目的としているため、商用利用でなければ卒業後も製品を使い続け ることが可能となっています。

Microsoft Azure を利用するために必要な知識

1. サーバー

複数のユーザーからのアクセスを受け付けるコンピューターのことをサーバーといいます。サーバーはアプリ から送信されたデータを受け取って、登録や検索などの何らかの処理を行ったり、アプリに対して処理結果を 返したりします。

2. データベース

様々なデータを格納するためのソフトウェアをデータベースといいます。今回はユーザーが回答したアンケー トデータや、ユーザーが SNS に書き込んだコメントなどを、データベースの中に入れて管理します。

3. テーブル

データベースは、テーブルというデータの入れ物が複数集まって構成されています。たとえば、ゲームを作る にはユーザー情報、アイテム情報、キャラクター情報、ステージ情報など、様々なデータが必要です。こうい った用途や役割の異なる複数の種類のデータを組み合わせて一つのアプリが作られています。そのためユーザ ー情報を格納するためのテーブル、アイテム情報を格納するためのテーブルといったように、格納するデータ の種類ごとにテーブルが必要となるのです。

今回のサンプルアプリでは、扱う情報の種類が少ないので一つのアプリにつきテーブルも一つとしています。



サーバー、データベース、テーブルの役割

Microsoft Azure for DreamSpark の利用開始手順

1. Microsoft アカウントの作成と DreamSpark への登録

「学生向け DreamSpark」の Web サイトにアクセスします。

https://www.dreamspark.com/Student/Default.aspx

画面右下に、5種類の登録方法の案内が記載されています。基本的には学校の先生の指示に従って登録を行っ てください。学生個人の利用で、自分が該当する登録方法がわからない場合は、5番目の「またはドキュメン トを供給することができます」を選択すると、学生証の写真を送ることで認証を行うことが可能です。

Microsoft Azure の利用準備



「アカウントの作成」ボタンをクリックして、まずは Microsoft アカウントの作成を行います。

	アカワノト音手物	1
X	ドレスまたは電話番号	
	8	
→ ++++++++++++++++++++++++++++++++++++	インしたままにする	
tythe	>	
PhONE	アクセスできない場合	
アカウント	ードを使ってサインイン	
セキュリティ情報を追加すると受信トレイの保護に役立ちます	22	から登録
	~	
Microsoft	t アカウントをお持ちでない場合 新規登録	

画面に表示される手順に従ってアカウントを作成し、DreamSpark ヘサインインします。

DreamSpark の 5 種類の認証方法のうち、いずれかを選択して登録を行ってください。学生証の写真を送信 する方法の場合、登録完了まで 2~3 日かかる場合があります。

また、登録方法によってはメールによる認証作業が必要となる場合があります。メールに記載された URL を クリックすることで登録が完了となります。

認証が完了すると、以下のような画面が表示されます。



「続行」をクリックして、DreamSparkのホーム画面に進んで下さい。

2. Microsoft Azure の利用開始登録

DreamSpark のホーム画面から、「製品のダウンロード」の「詳細情報」をクリックします。



ダウンロードページの中から、「Microsoft Azure for DreamSpark」を選択します。

			ホーム	学生 教育機関 Dre	amSpark について FAG
学生			概要 ソフト	ウェアカタログ サポー	-ト ようこそ 可奈子 ④
K Vis 無料で利 ^{学生なら新しい Visual} でダウシロードできま	ual Studi 川田できます I Studio Community 2015 *.	O ○ E無料 ④		Visual Studio	
reamSpark には なたの学校を通じてきらにン	必要なツールが フトウェアを得ることができ	揃っています			2
開発者およびデザイナー「	向けツール				
SFite Visual Studio Code	Sfile Visual Studio Community 2015	PHP TOOLS	25.92 Xamarin for Students	Bitte GitHub Education Student Developer Pack	TELL Desemble 2
Visual Studio Code	Visual Studio Community 2015	PHP Tools for Visual Studio	Xamarin for Students	GitHub Student Developer Pack	Windows Embedded 8.1 Industry Pro with
・ サーバーとアブリケーション 「「現 Microsoft Microsoft	R Server tr Window,	The second se		Rate on all	⊖
Microsoft Azure for DreamSpark	Microsoft R Server	SQL Server 2014	SQL Server 2014 Extensional Service Pack 1	Windows Server 2012	Windows Server 2012 R2 with Update
G 					•••••••••••••••••••••••••••••••••••••••
トレーニングと資格 MVA	Microsoft Certifications	Microsoft Press* eBook	Imagine Lup X	pluralsight _o	
Microsoft Virtual	マイクロンフト認定資格	MS Press の規料電子書	Microsoft Imagine Cup	Pluralsight	

Microsoft Azure for DreamSpark の説明画面に進んだら、「今すぐ登録」をクリックします。



利用者情報の入力画面が表示されますので、必要な情報を入力して Microsoft Azure へのサインアップを完了 してください。

完了すると、以下の画面が表示されます。



「サービスの管理を開始する」をクリックすると、Microsoft Azureの管理画面が表示されます。



この画面は「ダッシュボード」と呼ばれる、Azureの様々な機能を管理する画面です。サンプルアプリから 登録したデータもここで確認できます。

今後よく利用する画面になりますので、URLをブラウザのお気に入りやブックマークに登録しておきましょう。

3. データベースの作成

Azure のダッシュボードで右上の「新規」を選択し、「データ + ストレージ」「SQL Database」の順に選択 します。

Microsoft Azure の利用準備



※ 2017年1月現在、「データ + ストレージ」は「Databases」に名称が変更されています。

SQL Database の作成画面に進んだら、以下のように各項目を入力していきます。

「データベース名」に任意の名前を入力します。(①)

「サブスクリプション」はデフォルトの「DreamSpark」のままで OK です。

「リソースグループ」は「新規作成」ラジオボタンをチェックし、任意のグループ名を入力します。(2)

「ソースの選択」はデフォルトの「空のデータベース」のままでOKです。

「サーバー」	をクリッ	クして次のサー	バー作成手順に進みます。	(3)
--------	------	---------	--------------	-----

Microsoft Azure 🗸	新規 > データ + ストレージ > SQL Database	\mathcal{P} リソースの検索	× Q 🕸 😳 🧿 📖	อร่าเปปห
≡	SQL Database $ \square$ \times			
十 新規				
 ŵ リソースグループ # すべてのリソース ◎ 最近使うた項目 ③ App Service ◎ Virtual Machines ③ SQLデータペース Ŷ サブスクリプション 参照 > 	 ケラータペース名 monaca-database サブスクリプション DreamSpark リリース ガループ● ● 新規告載 ○ 既存のものを使用 Group1 ソースの選択● 空のテータペース シーメースの選択● 空のデータペース ・ ・ サーバー ・ ・ 第名の設定の構成 ・ 3 ・ ・ 細胞にぺい● S2 Standard ・ ・ 断合順序 ● SQL_Latin1_General_CP1_CLAS 			
	ダッシュボードにごン留めする			
	作 近			,

右側にサーバーを作成するためのパネルが表示されます。サーバーの作成手順は次の通りです。

「サーバー名」に任意の名前を入力します。この名前は世界中で重複することのない、唯一の名前にしなけれ ばなりません。(④)

「サーバー管理者ログイン」「パスワード」「パスワードの確認」はそれぞれ任意のものを入力して下さい。(⑤) 「場所」というのは、実際にサーバー本体が置かれている場所のことです。Azureのサーバーは世界中に配置 されていてどこのサーバーを使っても良いのですが、近い地域を選択した方が、通信速度が速くなる可能性が 高くなります。ここでは東日本または西日本を選択します。(⑥)

「Azure サービスにサーバーへのアクセスを許可する」はチェックが入ったままの状態で OK です。 上記すべてが入力できたら、一番下の「選択」ボタンをクリックします。(⑦)



右側のパネルが閉じて、「サーバー」と「価格レベル」の表示が以下のように変わります。 「ダッシュボードにピン留めする」にチェックを入れ(⑧)、「作成」ボタン(⑨)を押したら完成です。

Microsoft Azure の利用準備

Microsoft Azure 🗸	新規 〉 データ + ストレージ > SQL Database	<i>り</i> リソースの検索	×	Ĵ 🕸	\odot	?	
	SQL Database 🛛 🗕 🗖 🗡						
十 新規							
 ・リソース グループ ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	 * データペース名 monaca-database ✓ * サブスクリプション 						
🕒 最近使った項目	DreamSpark 🗸						
App Service	* リソース グループ ♥ ● 新規作成 ○ 既存のものを使用						
👰 Virtual Machines	Group1 🗸						
👼 SQL データペース	★ ソースの選択 ●						
📍 サブスクリプション	空のテータベース						
参照 >	* ^{サーバー} > > > > > > > > > > > > > > > > > > >						
	* 価格レベル @ > > Free > > * 価格レベル @ > + Mac * Mac						
	SQL_Latin1_General_CP1_CLAS						

4. App Service の作成

Microsoft Azure の「App Service」は、アプリ開発に必要な様々な機能を備えたサービスの総称です。デー タベースへの接続はもちろん、プッシュ通知、ユーザー認証、アプリ利用状況の分析など、非常に多くの機 能が含まれています。先ほど作成したデータベースへの接続も、App Service を通して行います。

まずは Azure のダッシュボードで右上の「新規」を選択し、「Web + モバイル」「Mobile App」の順に選択 します。



Microsoft Azure の利用準備

Mobile App の作成画面に進んだら、以下のように各項目を入力していきます。

「アプリ名」に任意の名前を入力します。世界中で重複することのない、唯一の名前にしてください。(①) 「サブスクリプション」はデフォルトの「DreamSpark」のままで OK です。

「リソースグループ」は「既存のものを使用」ラジオボタンをチェックし、ドロップダウンリストの中からデ ータベース作成の際に設定したグループ名を選択します。(②)

「App Service プラン/場所」>「新規作成」をクリックして右側のパネルを表示します。(③④)

「App Service プラン」にはどんな名前を付けても構いません。ここでは「ServicePlan」としています。(⑤) 「場所」は「Japan East」か「Japan West」にしましょう。(⑥)

「価格レベル」にはデフォルトで無料のプランが選択されていますので、そのままで構いません。

右下の「OK」ボタンをクリックして、右側のパネルを閉じます。(⑦)

「ダッシュボードにピン留めする」にチェックを入れ(⑧)、「作成」ボタン(⑨)を押したら完成です。



少し時間がかかりますが、App Service の作成が完了すると以下の画面が表示されます。 この画面にはダッシュボードからいつでもアクセスできます。

Microsoft Azure 🗸	monaca-app	در، حر	リースの検索	× 🖓 🏟 🙄 🕐	
≡	monaca-app		* _ □	×	
十 新規		☑ 参照 ■ 停止 ^ビ オ スワップ ひ 再	起動 面前除 ••••项目表示		
📦 リソース グループ	_>	要点 ^		A	
## すべてのリソース	🛄 概要	リソース グループ Group1	URL http://monaca-app.azurewebsites.net		
🕒 最近使った項目	監査ログ	Running	ServicePlan (Free)		
🔇 App Service	Access control (IAM)	Japan East	FTP/デブロイヌントユーザー セットがありません		
🧖 Virtual Machines	In 1913	DreamSpark	FTP AARS		
🥃 SQLデータペース	🗙 トラブルシューティング	menteer net whit have werden	The second product of the transme		
💡 サブスクリプション	APP SERVICE プラン		··· ×		
参照 >	📕 App Service ブラン	要求とエラー			
	🔀 スケール アップ (App Service の…	100			
	ピ スケール アウト (App Service の…	80			
	🛍 Quotas	60			
	📕 App Service プランの変更	40			
	リソース管理	使用でき 	きるデータはありません。		
	🗳 ዕイック スタート	0			
	₩ アプリケーション設定	16時 HTTP SERVER ERRORS REQUESTS	16:15 16:30 16:45	5	
		- -			
	🥔 バックアップ	17/1	ションを追加 (+)		
	カスタム ドメインおよび SSL *			•	

次に、作成した App Service と先ほど作成したデータベースを関連付けます。左側に表示されている一覧の中から「データ接続」を選択し、「Add」アイコンをクリックします。

Microsoft Azure 🗸	monaca-app - Data Connections	P リソースの検索		× 🗘 🎄 😳	0	
	monaca-app - Data Connection	ns	* _ 🗆 ×			
十 新規	+ Add					
📦 リソース グループ	▲ パフォーマンス テスト A	ТҮРЕ				
… すべてのリソース	(生) リソース エクスプローラー No Dat	ta Connections				
③ 最近使った項目	Zend Z-Ray					
	🌻 トラフィック ルーティング					
	■ 拡張機能					
Virtual Machines	モバイル					
<u>■</u> SQL データペース	🔚 Easy Tables					
📍 サブスクリプション	🐼 Easy API					
参照 >	💼 データ接続					
	📮 プッシュ					
	API					
	III API 定義					
	📮 CORS					
	MONITORING					
	✔ 豐告					
	🔸 診断ログ					
	🗖 ログストリーム					
	× インスタンスあたりのメトリックス (App… 🗸					



「SQL Database 必要な設定の構成」をクリックし、先ほど作成したデータベースを選択します。

次に「Connection string」をクリックします。

「Name」はデフォルトで値が設定されるのでそのままで構いません。

「User Name」と「Password」には先ほどデータベース作成の際に入力した「サーバー管理者ログイン」「パ スワード」と同じものを入力します。(①②)

最後に二か所の「OK」ボタンを押したら App Service の作成は完了です。(③④)

Image: state	Microsoft Azure 🗸 «	Add data connection > Connection string	り リソースの検索	× 🗘 🅸 😳 Ø	
 ★ fxQ ★ fxC 000-7.4 ★ fxC 000-7.4<!--</th--><th>= ×</th><th>\checkmark Add data conn\cdots \square \times</th><th>Connection string</th><th>_ 🗆 ×</th><th></th>	= ×	\checkmark Add data conn \cdots $ \square$ \times	Connection string	_ 🗆 ×	
Image: Py-2/h-7 Image: Py-2/h-7 <th>十 新規</th> <th></th> <th></th> <th></th> <th></th>	十 新規				
	 (i) リソース クループ III オペエのリメース ③ 高近使った項目 ③ App Service IVirtual Machines IVIrtual Machines IVIrtual Machines SQL データペース ? リプスクリプション 参照 > 	Type SQL Database SQL Database SQL Database Connector string MS_TableConnectionString	Name MS_TableConnectionString User Name monaca Password C	✓ ✓ ✓	
		ок (4)	ок 3		

最後に、App Service の詳細設定を変更します。Monaca から Azure に接続するために、マイクロソフトから 提供されている「MobileServices.Web.min.js」という JavaScript のプログラムを使いますが、本教材執筆時 点のバージョンを利用する場合、App Service 側で設定が必要になります。なお、今後プログラムのバージョ ンアップに伴って以下の手順は不要となる可能性があります。

右側の一覧の中から「アプリケーション設定」を選択し、設定画面を表示します。アプリ設定という項目の一 番下の行に「キー」「値」という入力欄がありますので、以下の文字列を入力します。

- \neq : $MS_SkipVersionCheck$
- 值:「true」

Microsoft Azure 🗸	monaca-app - アプリケーションの設定	り リソースの検索		× 🗘 🕸	0	
≡	📑 monaca-app - アプリケーシ	ョンの設定 *	_ 🗆 ×			
十 新規	monaca-app	□ 保存 × 破束				
🗊 リソース グループ		自動スワップ (7) オン	^			
… すべてのリソース	[] 概要	自動スワップ スロット	~			
🕒 最近使った項目	監査ログ	<i>∓</i> Iĭω <i>t</i> Ĩ				
🔇 App Service	Access control (IAM)	リモート デパッグ オン オン				
💻 Virtual Machines	🖉 91	リモート Visual Studio のバージョ				
🗟 SQLデータペース	🗙 トラブルシューティング	2010 2013 2015				
💡 サブスクリプション	APP SERVICE プラン	アブ設定				
参照 >	📕 App Service プラン	WEBSITE_NODE_DEFAU···· 4.2.3 Zロットの設定				
	ピ スケール アップ (App Service の…	MobileAppsManagement… latest Zロットの設定				
	ピ スケール アウト (App Service の…	- 値 20ットの設定				
	🛍 Quotas					
	📕 App Service プランの変更					
	リソース管理	IRのにスチアうい用いオトないにしょうていよう IRかにスチアうい用いない				
	4 クイック スタート	MS_TableConnect··· < セキュリティ上の理··· SQL Database スロットの設	定 …			
	三 アプリケーション設定	名前 値 SQL Datab… 🗸 🗌 スロットの設	de			
	∲ 認証/承認	0月 min の 11 m				
	💣 バックアップ	成正のトキユメント 				
	う カスタム ドメインおよび SSL ▼	Deraut.nun	•			

入力が完了したら「保存」アイコンを押して、変更を確定します。

Microsoft Azure 🗸	monaca-app - アプリケーションの設定	<i>P</i> リソースの検索	>	× 🗘 🏟 😳 🕐	
≡	monaca-app - アプリケー	ションの設定	* _ 🗆 ×		
十 新規	Thomaca app	日 保存 × 破策			
📦 リソース グループ		自動スワップ オン	^		
## すべてのリソース	🚺 概要	自動スワップスロット	~		
🕒 最近使った項目	監査ログ	デバッグ			
🔇 App Service	Access control (IAM)	リモート デバッグ オフ オン			
💻 Virtual Machines	🥏 91	リモート Visual Studio のバージョ			
🗟 SQLデータベース	🗙 トラブルシューティング	2017 2013 2015			
📍 サブスクリプション	APP SERVICE プラン	アブリ設定			
参照 >	📕 App Service プラン	WEBSITE_NODE_DEFAU···· 4.2.3	の設定 ・・・		
	🔀 スケール アップ (App Service の…	MobileAppsManagement… latest コスロットの	の設定 ・・・		
	🔀 スケール アウト (App Service の…	MS_SkipVersionCheck 🗸 true 20%	の設定 ・・・		
	🛍 Quotas	<i>キー</i> 値 20%M	の設定 ・・・		
	App Service プランの変更	ide (ar yr chr Tal			
	リソース管理	技称(スイチャ) 接続文字列の値が非表示になっています 接続文字列の値の表示			
	🗳 クイック スタート				
	Ξ アプリケーション設定	MS_TableConnect··· < セキュリティ上の理··· SQL Database	人山ットの設定 ···		
		名前 值 SQL Datab… V	スロットの設定		
	🥌 バックアップ	照定のドキュメント			
	カスタム ドメインおよび SSL ・	Provide the Control of			

5. テーブルの作成

右側の一覧の中から「Easy Tables」を選択すると、「Need to configure Easy Tables/Easy APIs – Click here to continue」と書かれている青い帯が表示されるので、クリックします。

Microsoft Azure 🗸	monaca-app - Easy Tables	<i>リ</i> ソースの検索		< 🗘	\$ \$	0	既定のディレクトリ
	monaca-app - Easy Tables		* _ 🗆 ×				
十 新規	+ AC	d 🚡 Add from CSV					
📦 リソース グループ	▲ パフォーマンス テスト	eed to configure Easy Tables/Easy APIs - Click he	ere to continue \rightarrow				
!!! すべてのリソース	(生) リソース エクスプローラー NAM	1					
④ 最近使った項目	Zend Z-Ray No re	sults.					
🔇 App Service	トラフィック ルーティング						
Virtual Machines	L. 拡張機能						
🗟 SQLデータベース	モバイル						
サブスクリプション	Easy Tables						
参照>	Easy API						
	 デージ技術 ブッシュ 						
	API						
	≧ API 走義						
	MONITORING						
	インスダンスめにリのメトリックス (App…						

「I acknowledge that this will overwrite all site contents.」(①) にチェックを入れて、「Initialize App」ボ タン(②) をクリックします。この作業が必要なのは最初の一回だけで、次にテーブルを作る際は不要な手 順となります。

Microsoft Azure 🗸	monaca-app - Easy Tables > Easy Tables	₽ リソースの検索	× 🗘 🏟 😳 🧿	
≡	- Easy Tables	* _ 🗆 ×	Easy Tables	_ 🗆 ×
十 新規	+ Add Add from CSV			
 ● リノース グループ ● ボムゼ クレープ ● ボムゼ クレース ● ボムゼ クレル目 ③ App Service ■ Virtual Machines ■ SQL データペース ● サブスクリチョン 参照 > 	Add A Add from C3V Need to configure Easy Tables/Easy APIs - Click he Noresults.	ere to continue →	Easy Tables is not supported on your current App Service app for Easy Tables support. Connect a database Tou already have a data connection Tou already have a data connec	e app. Please initialize s. Note that this will s.

少しの時間待機すると、青い帯のパネルが閉じます。これでテーブルを利用するための最初の準備は完了です。

続いてテーブルの作成作業を行います。

「Add」アイコンをクリックしてテーブルの作成画面を表示し、以下のように各項目を入力します。

「Name」にテーブル名を入力します。ここでは、「アンケートアプリ」で使用するテーブル名「surveyData」 を入力します。(①)

Name 以下の項目は、テーブルに対してユーザーが行うことが可能な操作を設定するためのものです。

- Insert permission …… データの追加
- Update permission ……… データの更新
- Delete permission ……… データの削除
- Read permission …… データの参照
- Undelete permission ……… データの復元

また、それぞれの選択肢の意味は以下の通りです。

- Allow anonymous access ………… 誰でも許可
- Authenticated access only ……… 認証されたユーザーのみ許可

「アンケートアプリ」では、ユーザーはアンケートに回答する機能だけを持ちます。またログイン機能は付け ずに誰でも回答できるようにしますので、以下のように設定します。(②)

- Update permission [Disabled]
- Delete permission …………………… 「Disabled」
- Undelete permission ………………… 「Disabled」



最後に「OK」ボタンを押すと、テーブルが作成されます。 つづいてテーブル名をクリックすると、テーブルの情報を確認することができます。

Microsoft Azure 🗸	monaca-app - Easy Tables > surv	eyData (0 records)		× L	? ⇔ ⊙ ⊙	
=	- Easy Tables		* _ 🗆 ×	surveyData (0 records)		_ 🗆 ×
十 新規	🕇 Add 🕌 Add fr	om CSV		🗙 Change permi… 🔀 Edit script	🗙 Manage schema 🛛 Refre	sh ••• 項目表示
🗊 リソース グループ	NAME			ID ^ CREATEDAT	UPDATEDAT VERSION	^ DELETED ^
# すべてのリソース	surveyData			結果がありません		
● 最近使った項目						
🔕 App Service						
👰 Virtual Machines						
🥃 SQL データペース						
💡 サブスクリプション						
参照 >						

「Manage schema」アイコンを押して、「Schema」というパネルを表示します。

Microsoft Azure 🗸	monaca-app -	Easy Tables > surveyData (0 records) > Schema	♀ リソースの検索		× 🖓 🏟 🙄 (?
≡	n x	surveyData (0 records)	_ 🗆 ×	Schema		_ 🗆 ×
十 新規		🗙 Change permi 🕫 Edit script 🔀 Manage schema 🕻	Refresh •••• 項目表示	- Add a column		
📦 リソース グループ		ID ^ CREATEDAT ^ UPDATEDAT ^ VERS	SION ^ DELETED ^	NAME	ТҮРЕ	IS INDEX
ま すべてのリソース		結果がありません		id	String	true …
⑤ 最近使った項目				createdAt	Date	true ····
🔕 App Service				updatedAt	Date	false
Virtual Machines				version	Version	false
👼 SQLデータベース				deleted	Boolean	false
 ↑ サブスクリプション						
9m /						
	1					

「Schema」というのは、テーブルの構造のことです。テーブル(表)はレコード(行)とカラム(列)によっ て構成されています。たとえばユーザー情報を格納するテーブルであれば、名前・年齢・性別・住所・電話番 号など、1ユーザーにつき様々な項目が必要です。そのとき、一件分のユーザーを表すのがレコード(行)、名 前や年齢などの項目を表すのがカラム(列)となります。 以下のようなイメージです。

					עכת	_
	名前	年齢	性別	住所	電話番号	
	最中 太郎	30	男性	東京都文京区	03-***-***	レコード
l	最中 次郎	25	男性	東京都杉並区	03-***-***	
ſ	温泉 花子	20	女性	千葉県千葉市	043-***-***	

ユーザー情報テーブル

テーブルを作成した時点で、あらかじめ最低限必要な5つのカラムが定義されています。

- id ………… レコードを識別するための唯一の値(レコード追加時に自動的に設定される)
- createdAt …… 作成日
- updatedAt …… 更新日
- version ………… バージョン(レコードを更新するたびに自動で設定される)
- deleted …………… 削除フラグ(削除されているときは true、削除されていないときは false)

「アンケートアプリ」では「年齢」「性別」「満足度」の 3 つの項目を収集しますので、それぞれに対応した 「name」「gender」「satisfaction」カラムを追加していきます。

Microsoft Azure v « surveyData (0 records) > Schema > Add a column 🗖 × Schema Add a column ┿ 新規 🕂 Add a column 📦 リソース グループ IS INDEX * Column name TYPE NAME name 👖 すべてのリソース id String true Data type String createdAt Date ~ 🕒 最近使った項目 true updatedAt Date false Service version Version false 👰 Virtual Machines deleted false Boolear 👼 SOI データベース 💡 サブスクリプション 参照 >

「Add a column」アイコンを押して、カラム追加画面を表示してください。

「Column name」にはカラム名を入力します。まずは名前を格納する項目である「name」を入力します。 「Data type」はデータの種類を設定する項目です。以下の4つの中から選択します。

- String …… 文字列
- Number …… 数值
- Date …… 日付

• Boolean ……… 論理値(true または false)

名前は文字列型なので、「String」を選択します。最後に「OK」ボタンを押したらカラムの追加は完了です。

同様の手順で、「gender」カラムと「satisfaction」カラムも追加してください。どちらも Data type は「String」 にしてください。

Microsoft Azure 🗸	monaca-app -	Easy Tables > surveyD	Data (0 records) > Schema	-עני סק	-スの検索
		Schema			_ 🗆 ×
十 新規	項目表示	+ Add a column			
📦 リソース グループ	• ^	NAME	ТҮРЕ	IS INDEX	
罪 すべてのリソース		id	String	true	
● 最近使った項目		createdAt	Date	true	
🔕 App Service		updatedAt	Date	false	
Virtual Machines		version	Version	false	
👼 SQLデータベース		deleted	Boolean	false	
サブスクリブション		name	String	false	
42.872 \		gender	String	false	
9m /		satisfaction	String	false	
	4				

すべて完了すると以下のような画面になります。

以上で、Azure 側の準備は完了です。続いては、Azure にデータを保存するサンプルアプリの作成に進みます。

サンプルアプリの紹介

アンケートアプリ

ユーザーが回答したアンケートを収集するアプリです。アンケートデータは Microsoft Azure 上に蓄積され、 Web の管理画面で閲覧できます。このアプリを題材として、アプリから Azure 上のデータベースへデータを 登録する方法を学びます。



プロジェクトの作成



まずは、インポートした状態のプロジェクトについて解説します。



js フォルダの中に、以下のファイルが配置されています。 MobileServices.Web.js

この JavaScript ファイルはマイクロソフトから提供され ているもので、Azure のデータベースを操作するための命 令が含まれています。

index.html 10 行目の<script>タグでこのファイルが読み 込まれています。

index.html

25	<body></body>
26	<h1>アンケート</h1>
27	<div id="surveyForm"></div>
28	<h2>お名前</h2>
29	<input id="name" type="text"/>
30	<h2>性別</h2>
31	<select id="gender"></select>
32	<option value="man">男性</option>
33	<option value="woman">女性</option>
34	
35	<h2>満足度</h2>
36	<select id="satisfaction"></select>
37	<option value="1">とても良い</option>
38	<option value="2">良い</option>
39	<option value="3">普通</option>
40	<pre><option value="4">悪い</option></pre>
41	<option value="5">とても悪い</option>
42	
43	
44	<hr/>
45	<button id="sendBtn" onclick="send()">送信</button>
46	

<body>タグの中には、アンケートフォームを定義するタグが記述されています。<input>、<select>、<option>
タグについての詳細は第9章を確認してください。

```
<script>
11
           // Azureの接続準備
12
          var azure = new WindowsAzure.MobileServiceClient(
13
               'App Service 𝒯 URL'
14
           );
15
          var table = azure.getTable("surveyData");
16
17
18
          // 送信ボタンを押したときの処理
19
          function send() {
20
21
           }
22
      </script>
23
```

<script>タグには、あらかじめプログラムが穴埋めで記述されています。 それでは、このプログラムを埋めてアンケートアプリを完成させましょう。

実習

13 行目から 15 行目に記述されているのが Azure へ接続するための命令です。「App Service の URL」と記述 されている箇所を書き換えましょう。URL は Azure のダッシュボードから確認できます。

| 13 | <pre>var azure = new WindowsAzure.MobileServiceClient(</pre> |
|--------------------|---|
| 14 | 'App ServiceのURL' |
| 15 |); |
| | |
| Microsoft Azure 🗸 | ^{アルイズの} 自分の Azure 環境の ^{© ©} |
| ≡ | in monaca-app
ER(ルック) |
| 十新規 | |
| 📦 リソース グループ | ○ 成度(Cttl+/) ● 要点 ▲ |
| … すべてのリソース | リソース グループ
Group1
・
は思
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
・
11日
11日 |
| 🕒 最近使った項目 | マロック App Server 57/met/マル
室 監査ログ Running ServicePan (Free)
単美 ServicePan (Free) |
| 🔇 App Service | amm Amm FTP / プロパント エッシーム
Amm Japan East FTP / プロイ ユーザー セットがありません
サログロドゥンタ FTP より A |
| 👰 Virtual Machines | yy DreamSpark TTS #245 |
| 🥃 SQL データペース | X 1577021-5707 attat teal teal teal teal teal teal tea |
| 💡 サブスクリプション | APP SERVICE 75> ···· × |
| 参照 > | L App Service プラン 要求とエラー |
| | 区 スケール アップ (App Service の… 100 |
| | 図 スケール アウト (App Service の… 80 80 |
| | iii Quotas 60 60 |
| | App Service プランの変更 40 40 40 40 40 40 40 40 40 40 40 40 40 |
| | 使用できるデータはありません。 リソース並 理 |
| | ۵ |
| | ■ アプリケーション設定
IntTP SERVER BROKS ■ REQUESTS |
| | Ŷ 153E/₩38 |
| | |
| | |

var table = azure.getTable("surveyData");

16 行目では、データを登録するテーブルを選択しています。ここでは先ほど作成した「surveyData」テーブ ルを指定しています。azure.getTable 命令が実行されると「テーブル情報」が返却され、table 変数の中に格 納されます。

続いて、送信ボタンを押したときに実行される send 関数の中に処理を追記します。

19	// 送信ボタンを押したときの処理
20	<pre>function send() {</pre>
21	
22	}

```
// 送信ボタンを押したときの処理
function send() {
    var item = {
        name: document.getElementById("name").value,
        gender: document.getElementById("gender").value,
        satisfaction: document.getElementById("satisfaction").value
    };
    // Azure にデータ登録
    table.insert(item)
    .done(function () {
        alert("登録が完了しました");
    }, function (err) {
        alert('error:' + err);
    });
}
```

まず、アンケートフォームに入力された内容を取得し、まとめて item 変数の中に代入しています。 つづけてデータの登録を行うための「insert」命令を使って、item 変数の内容を Azure に送信しています。

データの登録

16

テーブル情報.insert(登録するデータ).done(登録成功時に実行する関数,登録失敗時に実行する関数);

サンプルアプリの紹介

insert 命令はデータの登録処理と、登録が終わった後に何をするのかをすべてまとめて指定するので、一行の コードが長くなりがちです。そういった場合はコードの途中で改行を行うことで見やすくなります。



以上で、プログラムは完成です。

実行結果

このアプリはプレビューでは実行させることができないので、Monaca デバッガーを使って実行しましょう。



「登録が完了しました」というメッセージが表示されたら、回答データが Azure のデータベースに登録された ことを確認してみましょう。Azure のダッシュボードを開き、App Service の一覧の中から「Easy Tables」 「surveyData」の順に選択すると、surveyData テーブルに登録されているデータの一覧が表示されます。 1件分のデータが登録されていれば成功です。

サンプルアプリの紹介

Microsoft Azure 🗸	monaca-app - E	asy Tables 🗲 surveyData (1 record)	₽ リソースの検索	× 🗘 🏟 😳 Ø	
	- Easy Table	25	* _ 🗆 ×	surveyData (1 record)	_ (□)×
十 新規		+ Add 🛧 Add from CSV		🗙 Change permi ··· 🕅 Edit script 🗙 Manage schema	ひ Refresh ・・・・項目表示
📦 リソース グループ	^	NAME		ID ^ CRE ^ UPD ^ VER ^ DEL ^ N	IAME ^ GEN ^ SATL ^
₩ すべてのリソース		surveyData		✓ 4b147… 2016… 2016… AAAAA… false #	最中 太郎 man 1
🕒 最近使った項目					
🔇 App Service					
👰 Virtual Machines	/レビュー)				
👼 SQL データペース					
💡 サブスクリプション	- L				
参照 >					

右上の最大化アイコンを押すと、以下のように横幅が広がって見やすくなります。更新するときは「Refresh」 アイコンを押します。

Microsoft Azure 🗸	monaca-app - Easy Tables > surveyData (1 record) 🛛 🏸 リンースの検索 🛛 🗘 😨 😨 😨 😰 🕲 🕲 🕲 🕲 🕲 🕲 🖉
=	surveyData (1 record) _ 🗗 ×
十 新規	🗙 Change permi 🕫 Edit script 💥 Manage scherna 🕐 Refresh 🝵 Delete table 🗶 Clear table 🍵 Delete rows 🕤 Strearning logs
🗊 リソース グループ	ID ^ CREATEDAT ^ UPDATEDAT ^ VERSION ^ DELETED ^ NAME ^ GENDER ^ SATISFACTION ^
… すべてのリソース	4b147503-a40f-43ac… 2016-08-16T09:18:1… 2016-08-16T09:18:1… AAAAAAAB91= false 最中太節 man 1
🕒 最近使った項目	
🔇 App Service	
Virtual Machines	
🥫 SQL データベース	
💡 サブスクリプション	
参照 >	

なお、完成版のソースコードは以下に公開されています。うまく動かない場合は参考にしてください。

https://ja.monaca.io/book/001/azure-01z.zip

匿名 SNS

複数のユーザーが匿名でコメントを書き込み、コミュニケーションを行う SNS アプリです。自分が書き込ん だコメントは他のユーザーのコメントとは異なる色で表示され、コメントの編集・削除ができます。このアプ リを題材として、Azure 上のデータベースにデータを登録・参照・更新・削除する方法を学びます。

Monaca SNS 🛛 😋 🕞	く Back コメント編集
やばい遅刻しそう	運動的ロだと
課題明日だよ- >	11月11日によー
今日って課題の提出あったっけ?	
雨降ってるし(。。)	
おはようございます!	保存
おはよー	

なお、画面の構築には Onsen UI を利用しています。これらの技術については本教材では詳しく説明を行いま せんので、別途 Onsen UI の教材を確認してください。 https://ja.monaca.io/book/001/onsen-01.pdf

プロジェクトの作成



テーブルの準備

15~18ページを参考に、以下の内容でテーブルを作成します。

Name : [userComments]

permission :

•

- - Update permission [Allow anonymous access]



Schema

Column Name : Comment	t」 /	∫ Data Type : 「String」
Column Name : 「uuid」	/	´ Data Type : 「String」

Microsoft Azure 🗸 🖛	onaca-app - Easy Tables 🛛	> userComments (4 recor	rds) > Schema		Q	(]	3 😳	?	
≡	Schema			∎ ×					
十 新規	🕂 Add a column								
📦 リソース グループ	NAME	ТҮРЕ	IS INDEX						
… すべてのリソース	id	String	true						
④ 最近使った項目	createdAt	Date	true						
🔇 App Service	updatedAt	Date	false						
👰 Virtual Machines	version	Version	false						
🧕 SQL データペース	deleted	Boolean	false						
📍 サブスクリプション	comment	String	false						
参照し	uuid	String	false						
				_					

プロジェクトの構成

プロジェクトは以下の構成になっています。プログラムは js/app.js 内に記述していきます。



index.html (TOP ページ)

18	<body></body>
19	モーダル画面
20	<ons-modal id="modal"></ons-modal>
21	<pre><ons-icon icon="spinner" size="50px" spin="true"></ons-icon></pre>
22	
23	
24	ナビゲーター
25	<ons-navigator id="navi" page="list.html"></ons-navigator>
26	

最初に表示される index.html では、モーダル画面(待機中に表示されるグレーの画面)と、ナビゲーター(Onsen UI で複数画面を管理するタグ)が定義されています。 ナビゲーターにより、list.html を読み込んで表示しています。

list.html(一覧画面)

1	<pre><ons-page id="list-page"></ons-page></pre>
2	<ons-toolbar></ons-toolbar>
3	<pre><div class="center">Monaca SNS</div></pre>
4	<pre><div class="right"></div></pre>
5	<pre><ons-toolbar-button class="add-btn" onclick="showNewPage()"></ons-toolbar-button></pre>
6	<pre><ons-icon icon="ion-chatbubble-working" size="28px"></ons-icon></pre>
7	
8	<pre><ons-toolbar-button class="refresh-btn" onclick="showComments()"></ons-toolbar-button></pre>
9	<pre><ons-icon icon="ion-refresh" size="28px"></ons-icon></pre>
10	
11	
12	
13	
14	<ons-list id="timeline"></ons-list>
15	
16	

list.html は、ユーザーによって書き込まれたコメントの一覧を表示する画面です。14 行目~15 行目の<onslist>タグ内にコメント一覧を表示します。

ツールバー上には追加ボタン(吹き出しのアイコン)と更新ボタン(丸い矢印のアイコン)が定義され、追加 ボタンを押したときには「showNewPage」関数が、更新ボタンを押したときには「showComment」関数が実 行されます。

new.html(追加画面)

```
<ons-page id="new-page">
1
       <ons-toolbar>
2
          <div class="left"><ons-back-button>Back</ons-back-button></div>
3
          <div class="center">コメント追加</div>
4
       </ons-toolbar>
5
6
       <div class="input-form">
7
           <textarea id="add-comment" class="textarea" placeholder="コメントを入力"></textarea>
8
           <ons-button id="add-save-btn" class="button" onclick="addComment()">
9
               保存
10
          </ons-button>
11
       </div>
12
  </ons-page>
13
```

new.html は、コメントを追加する画面です。コメントの入力欄と、保存ボタンが表示されます。保存ボタン を押すと、「addComment」関数が実行されます。

edit.html (編集・削除画面)

```
<ons-page id="edit-page">
1
       <ons-toolbar>
2
           <div class="left"><ons-back-button>Back</ons-back-button></div>
3
           <div class="center">コメント編集</div>
4
           <div class="right">
               <ons-toolbar-button id="del-btn">
6
                   <ons-icon icon="ion-trash-b" size="28px"></ons-icon>
7
               </ons-toolbar-button>
8
           </div>
9
       </ons-toolbar>
10
       <div class="input-form">
12
           <textarea id="edit-comment" class="textarea" placeholder="本文"></textarea>
13
           <ons-button id="edit-save-btn" class="button">
14
               保存
15
           </ons-button>
16
       </div>
17
  </ons-page>
18
```

edit.html は、コメントの編集と削除を行う画面です。保存ボタンを押すとコメントの編集、ツールバー上の 削除ボタン (ゴミ箱のアイコン)を押すと削除を行います。ボタンを押したときのイベント登録は js/app.js 内 で行います。

app.js

今回のプログラムは100行以上に及ぶ比較的長い処理ですので、app.js というファイルにまとめます。 以下の重要な処理のみ穴埋め形式になっていますので、順を追って処理を追加していきます。

(1)	起動時の処理
(2)	コメント取得処理
(3)	コメント追加処理
(4)	コメント編集処理
(5)	コメント削除処理

実習

(1) 起動時の処理

1	// (1) 起動時の処理
2	var table;
3 -	<pre>ons.ready(function() {</pre>
4	// Azureの接続準備
5	<pre>var azure = new WindowsAzure.MobileServiceClient("<u>App ServiceのURL</u>");</pre>
6	<pre>table = azure.getTable("userComments");</pre>
7	showComments();
8	}):

まずは、「App Service の URL」と記述されている箇所を書き換えます。自分の Azure 環境の URL をコピー して、貼り付けてください。

Azure への接続が完了すると、「showComments」関数を呼び出してコメント一覧の表示を行います。

(2) コメント取得処理

以下のコードを追記して、「showComments」関数を完成させます。

```
// (2) コメント取得処理
function showComments() {
    document.getElementById("modal").show();
    // リストを初期化
    var list = document.getElementById("timeline");
    var htmlStr = '';
```

```
// 直近のコメントから、100件分取得する
table
.orderByDescending("createdAt")
                             // 更新日が新しい順に並び替え
                                  // 最大 100 件
.take(100)
.read()
.done(function (items) {
   // 1件ずつリストに追加する
   items.forEach(function (item) {
       htmlStr += createListItem(item);
   });
   list.innerHTML = htmlStr;
   document.getElementById("modal").hide();
}, function (err) {
   alert(err.message);
   document.getElementById("modal").hide();
});
```

まず、Azureからのデータの取得には少々時間がかかりますので、モーダル画面を表示して待機状態にします。 そしてコメントー覧を表示する<ons-list>タグ内をクリアして、空の状態にしたら準備完了です。

Azure からのデータの取得は、「read」という命令を使います。「read」の前にはデータの取得条件等を指定す ることが可能です。

データの取得

}

テーブル情報.取得条件.read().done(取得成功時に実行する関数,取得失敗時に実行する関数);

取得成功時に実行する関数の中では、「createListItem」関数を呼び出しています。この関数はあらかじめコー ドが埋められています。

Azure から取得した一件分のデータを受け取り、<ons-list-item>タグを生成して返却しています。<ons-listitem>タグ生成の際に、自分の書き込みであった場合(端末固有の ID である UUID を見て判定)は文字色を 青色に変更し、タップされた際に編集・削除画面に遷移するイベントの設定を行っています。

```
16 // 1 件分のメモを表すHTML要素を作成
  function createListItem(data) {
17 -
     var style = "";
18
     var clickEvent = "";
20
      // 自分の書き込みだった場合
      if(data.uuid == device.uuid) {
         style = "color:blue";
23
         clickEvent = "onclick='showEditPage(" + JSON.stringify(data) + ")'";
24
      }
26
      var item = "<ons-list-item modifier='longdivider' class='item' " + clickEvent + ">"
             + "" + data.comment + ""
28
             + "</ons-list-item>";
29
30
      return item;
31
  }
```

すべてのデータの挿入が終わったら、モーダル画面を非表示にして処理終了です。



アプリの初期起動時はコメントが登録されていない状態ですから、何も表示されません。つづいて一覧画面上 部のツールバーにある追加ボタン(吹き出しのアイコン)を押したときの処理を追加して、コメントの登録を 行えるようにします。

なお、追加ボタンを押したときに new.html(追加画面)に遷移する処理はあらかじめ記述されています。

```
33 // 追加ボタンを押したときの処理
34、function showNewPage() {
35 document.getElementById("navi").pushPage("new.html");
36 }
```

(3) コメント追加処理

以下のコードを追記して、「addComment」関数を完成させます。

```
// (3) コメント追加処理
```

function addComment() {

// 入力されたコメントと UUID を取得

```
var item = {
    comment: document.getElementById("add-comment").value,
    uuid: device.uuid
  };
  // Azure にデー夕登録
  table.insert(item)
  .done(function () {
      showComments();
      document.getElementById("navi").popPage();
  }, function (err) {
      alert(err.message);
  });
}
```

上記は追加画面で保存ボタンが押された際の処理です。Azureのテーブルには、入力されたコメント(comment カラム)と端末固有の ID である UUID(uuid カラム)の2つの情報を1件分のデータとして登録します。

データの登録

テーブル情報.insert(登録するデータ).done(登録成功時に実行する関数,登録失敗時に実行する関数);

登録が完了すると、一覧画面に戻り登録したデータが表示されます。他の端末からもデータを登録して、複数 の端末からのデータの書き込みが表示されることを確認しましょう。ツールバー上の更新ボタン(丸い矢印の アイコン)を押すと最新の情報が取得できます。



自分が登録したデータのみが、青色で表示されます。青色のコメントをタップすると、編集・削除画面に遷 移します。

サンプルアプリの紹介

iPhoneから登録	iPhoneから登録 保存	< Back	コメント編集	Î
	保存	iPhoneか	6登録	
	保存			
保存			保存	

編集・削除画面で保存ボタンおよび削除アイコンをタップしたときのイベント処理はあらかじめ記述されて います。それぞれ関数を呼び出す際にデータの ID(Azure のテーブルの id カラムの値)が引数として渡さ れている点が重要なポイントです。

```
44 // 編集ボタンを押したときの処理
45 -
  function showEditPage(data) {
      document.getElementById("navi").pushPage("edit.html")
46
      .then(function() {
47
          document.getElementById("edit-comment").value = data.comment;
48
49
          // 保存ボタンを押した時の処理
50
          document.getElementById("edit-save-btn").onclick = function() {
51
              editComment(data.id);
          };
54
          // 削除アイコンを押した時の処理
55
          document.getElementById("del-btn").onclick = function() {
56
              delComment(data.id);
57
58
          };
59
      });
60
  }
```

(4) コメント編集処理

以下のコードを追記して、「editComment」関数を完成させます。

```
// (4) コメント編集処理
function editComment(id) {
    var item = {
        id: id,
        comment: document.getElementById("edit-comment").value
    };
    // Azure のデータ更新
    table.update(item)
```

```
.done(function () {
    showComments();
    // 一覧ページに戻る
    document.getElementById("navi").popPage();
}, function (err) {
    alert(err.message);
});
}
```

上記は編集・削除画面で保存ボタンが押された際の処理です。Azureのテーブル内の既存データを編集する際は、必ず id カラムの値を指定して処理を実行します。対応する id のデータに対して上書き更新が行われます。

データの更新



(5) コメント削除処理

以下のコードを追記して、「delComment」関数を完成させます。

```
// (5) コメント削除処理
function delComment(id) {
    ons.notification.confirm({
        title: "確認",
        message: "このコメントを削除してよろしいですか?",
        callback: function (index) {
```

```
// キャンセルを押されたら終了
if (!index) return;
var item = { id: id };
// Azure のデータ削除
table.del(item)
.done(function () {
    showComments();
    // 一覧ページに戻る
    document.getElementById("navi").popPage();
    }, function (err) {
        alert(err.message);
    });
    };
}
```

上記は編集・削除画面でツールバー上の削除アイコンが押された際の処理です。編集処理と同様に、必ず id カ ラムの値を指定して処理を実行します。対応する id のデータが Azure のテーブルから削除されます。

データの削除



以上で完成です。

サンプルアプリの紹介

完成版のソースコードは以下に公開されています。うまく動かない場合は参考にしてください。

https://ja.monaca.io/book/001/azure-02z.zip