モバイルアプリを Azure で作る - 【開発環境を整える】

コマンドでのパッケージと実行準備

■テキストエディタとコマンドで開発する

Windows Azure Platform 上で動かす Web アプリケーションの開発にあたっては、Microsoft Visual Studio や Eclipse などの統合開発環境を使った開発が一般的ですが、使い慣れたテキストエディタ（秀丸など）を利用して開発することもできます。

使用するエディタに関しては、UTF-8 の文字コードで保存が可能で、なおかつ BOM （Byte Order Mark)が自動的に挿入されないものをお勧めします。スタートアップタスクに BOM がある場合、スタートアップタスクが実行されない可能性があるからです。本ガイドでは全て秀丸エディタを使用してテキストの編集を行っています。

・秀丸

http://hide.maruo.co.jp/software/hidemaru.html

■Azure 用 PHP Web アプリケーションを作成する前の準備

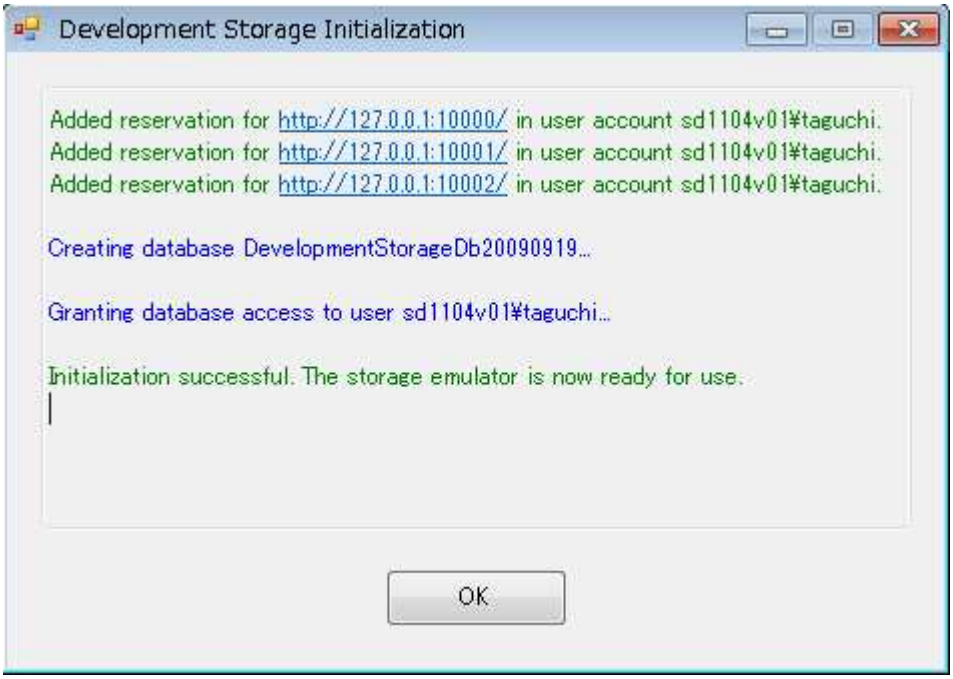
今回は、指定したファイルを Blob ストレージに保存するだけの簡単な PHP Web アプリケーションを開発します。

Blob ストレージへのアクセスには、Windows Azure SDK for PHP を使用します。Windows Azure SDK for PHP の入手と設定方法は、「Azure と PHP を使う準備」という章で紹介してますので、参考にしてください。

今回は、まずローカルで通常の IIS を使って開発を始めます。その後、Windows Azure 本番環境へアップロードできるように、コマンドを使って手動でパッケージを作成し、Windows Azure 本番環境にデプロイします。このために、詳しくは後述する CSPack と CSRun というコマンドを入手して使います。

開発するアプリケーションは、ローカルで実行するときに Storage Emulator を使用するので、事前に起動しておきます。Storage Emulator の起動は、Windows Azure SDK をインストールしてあれば、スタートメニューから行うことできます。

初回起動時は、[Development Storage Initialization]画面が表示され、Storage Emulator の初期化が行われます。



（Storage Emulator の初期化画面）

■Web ページの作成

まず、ファイルを選択するページの index.htm を wwwroot フォルダの直下に追加します。今回は、以下のような内容で作成します。

[index.htm]

```
<head>
    <title>ファイルアップロード</title>
</head>
<body>
    <h1>Windows Azure サンプル</h1>
    <form method="post" enctype="multipart/form-data" action="upload.php">
        <input name="file" type="file" /><br />
          <br />
        <input type="submit" value="アップロード" />
    </form>
</body>
</html>
```

次に、ファイルを Blob に保存するページとなる upload.php を追加します。ファイルの保存先として Storage Emulator を使い、動作確認をします。以下のような内容で作成します。

[upload.php]

```
<?php
require_once 'Microsoft/WindowsAzure/Storage/Blob.php';

$msg = 'ファイルをアップロードできません。';

if (is_uploaded_file($_FILES['file']['tmp_name'])) {
	$host = Microsoft_WindowsAzure_Storage::URL_DEV_BLOB;
	$accountName = 'devstoreaccount1';
	$accountKey   =
'Eby8vdM02xNOcqFlqUwJPLlmEtlCDXJ1OUzFT50uSRZ6IFsuFq2UVErCz4I6tq/K1SZFPTOtr/KBHBeksoGMGw==';
	$usePathStyleUri = true;

	$blobContainerName = 'filecontainer';
	$retryPolicy = Microsoft_WindowsAzure_RetryPolicy_RetryN::retryN(3, 100);
	$blobStorageClient = new Microsoft_WindowsAzure_Storage_Blob($host, $accountName,
$accountKey, $usePathStyleUri, $retryPolicy);

	if (!$blobStorageClient->containerExists($blobContainerName)) {
		$blobContainer = $blobStorageClient->createContainer($blobContainerName);
		$blobStorageClient->setContainerAcl($blobContainerName,
Microsoft_WindowsAzure_Storage_Blob::ACL_PUBLIC);
	} else {
		$blobContainer = $blobStorageClient->getContainer($blobContainerName);
	}

	$fileName = $_FILES['file']['name'];
	$result = $blobStorageClient->putBlob($blobContainerName, $fileName,
$_FILES['file']['tmp_name']);
	$msg = " $fileName ファイルをアップロードしました。";
}
?>
<html>
<head>
<title>ファイルアップロード</title>
</head>
<body>
<?php
  echo $msg;
?>
</body>
</html>
```

以上の作業で終了したら、さっそくブラウザを起動し、Web サイト（通常は http://localhost/ ）にアクセスしてください。

（Compute Emulator での実行画面（ブラウザー））

ページが表示されたら、適当なファイルを指定して「アップロード」ボタンをクリックすれば、ページが遷移しファイルの登録が完了します。

（Compute Emulator での実行結果画面（ブラウザー））

アップロードしたファイルは、CloudXplorer や Azure Storage Explorer などのアプリケーションで確認できます。

CloudXplorer
http://clumsyleaf.com/products/cloudxplorer

Azure Storage Explorer
http://azurestorageexplorer.codeplex.com/

■開発ファブリック上での動作確認

本番へ展開する前に、ローカルのエミュレータである Compute Emulator で、アプリケーションの動作確認をしてみましょう。

その前に、パッケージを作成するのに必要なサービス定義ファイルと、実行するために必要なサービス構成設定ファイルを作成する必要があります。作成したファイルは、wwwroot フォルダの親フォルダ（通常は C:¥inetpub、以下 inetpub）に追加します。

まずは、サービス定義ファイル（今回は、ServiceDefinition.csdef)を作成します。以下のような内容で、エンドポイントの設定を定義して作成します。

[ServiceDefinition.csdef]

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="PHPSample"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition">
  <WebRole name="WebRole" enableNativeCodeExecution="true">
    <Sites>
      <Site name="Web">
        <Bindings>
          <Binding name="Endpoint1" endpointName="Endpoint1" />
        </Bindings>
      </Site>
    </Sites>
    <Endpoints>
      <InputEndpoint name="Endpoint1" protocol="http" port="80" />
    </Endpoints>
  </WebRole>
</ServiceDefinition>
```

参考：Windows Azure Service Definition Schema

http://msdn.microsoft.com/en-us/library/ee758711

次に、サービス構成設定ファイル（今回は、ServiceConfiguration.cscfg）を作成します。今回は、以下のような内容で、Guest OS には 2.x 系を設定します。

[ServiceConfiguration.cscfg]

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceConfiguration serviceName="PHPSample"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceConfiguration" osFamily="2"
osVersion="*">
  <Role name="WebRole">
    <Instances count="1" />
  </Role>
</ServiceConfiguration>
```

参考：Service Configuration Schema
http://msdn.microsoft.com/en-us/library/ee758710

■CSpack と CSRun を使ったパッケージ化と実行

ファイルの作成が終了したら、Compute Emulator を使って、アプリケーションを動作させます。Compute Emulator や Windows Azure 本番環境で実行するには、CSPack を使ってサービスのパッケージを作成する必要があります。CSPack で作成したエミュレータ用のパッケージは、CSRun を使って実行可能です。

CSPack Command-Line Tool
http://msdn.microsoft.com/en-us/library/gg432988.aspx

CSRun Command-Line Tool
http://msdn.microsoft.com/en-us/library/gg433001.aspx

では、エミュレータ用のパッケージを作成します。「Windows Azure SDK Command Prompt」を実行（inetpub フォルダの UAC が有効の場合は、管理者として実行）し、inetpub フォルダに移動します。まず、以下のコマンドを実行し Compute Emulator で動作させるパッケージを作成します。

```
cspack ServiceDefinition.csdef /copyonly /out:SamplePackage.csx
/sitePhysicalDirectories:WebRole;Web;wwwroot /role:WebRole;wwwroot
```

「SamplePackage.csx」フォルダが作成されパッケージの作成が終了したら、以下のコマンドを実行します。問題がなければパッケージされたサービスが Compute Emulator 上で起動し、ブラウザが起動されます。

csrun SamplePackage.csx ServiceConfiguration.cscfg /launchbrowser

ブラウザが起動したら、先ほどと同様に適当なファイルを指定して「アップロード」ボタンをクリックすれば、ページが遷移しファイルの登録が完了します。

確認が終わったらブラウザを終了します。また、稼働している Compute Emulator を停止して削除するため、以下のコマンドを実行します。

csrun.exe /removeall

この後、本番環境へ展開します。Windows Azure ホストサービスと Windows Azure ストレージ サービスを使用するので、Windows Azure デベロッパー ポータルでホストサービスとストレージアカウントを事前に作成する必要があります。

■Windows Azure 本番環境への展開準備

ローカルでの確認が終わったら、本番環境へ展開していきます。その前に少しだけ修正をします。

ローカルと同様に、本番環境でも PHP Web アプリケーションを動かす場合には、PHP をインストールして環境を設定する必要があります。WebPI を用いてインストールすることも可能ですが、今回は本番と開発環境で PHP のバージョンに差が出ないように、PHP ランタイムを Web アプリケーションと一緒にパッケージ化します。スタートアップタスクで、FastCGI とハンドラマッピングの設定を行います。

最初に、PHP ランタイムを含めたパッケージを作成するため、Web サイト内にシンボリックリンクを作成します。

シンボリックリンク作成コマンド例

mklink /d "c:¥inetpub¥wwwroot¥php" "C:¥Program Files¥PHP¥v5.3"

次に、PHP アプリケーションを実行するための設定を、スタートアップタスクで行います。まず、スタートアップタスクで実行されるバッチとして setup.cmd を作成します。setup.cmd では、PowerShell の実行ポリシーを変更し、script.ps1 を実行します。今回は、以下のような内容で作成し、wwwroot フォルダの直下に bin フォルダを作成して追加します。

[setup.cmd]

```
@echo off
powershell -command Set-ExecutionPolicy RemoteSigned
powershell .¥script.ps1 "¥"%RoleRoot%¥""
```

次に、setup.cmd から実行される PowerShell スクリプト として script.ps1 を作成します。script.ps1 では、FastCGI とハンドラマッピングの設定を行います。今回は、以下のような内容で作成し、bin フォルダに追加します。

[script.ps1]

```
Import-Module Webadministration

$phpPath = $Args[0] + "¥approot¥php¥php-cgi.exe"
$basePath = $Args[0] + "¥base¥x86"

New-WebHandler -Name "PHP Handler" -Path *.php -Verb * -Modules FastCgiModule -ScriptProcessor $phpPath
-PSPath "IIS:¥Sites"
Add-WebConfiguration "/system.webServer/fastCgi" -value @{fullPath=$phpPath}
Add-WebConfiguration "/system.webServer/fastCgi/application/environmentVariables" -value
@{name='PATH';value=$basePath}
```

次に、サービス構成設定ファイル（ServiceDefinition.csdef)を修正して、スタートアップタスクとして setup.cmd を実行するようにします。以下のような内容になります。

[ServiceDefinition.csdef]

```
<?xml version="1.0" encoding="utf-8"?>
<ServiceDefinition name="PHPSample"
xmlns="http://schemas.microsoft.com/ServiceHosting/2008/10/ServiceDefinition">
  <WebRole name="WebRole" enableNativeCodeExecution="true">
    <Sites>
      <Site name="Web">
        <Bindings>
          <Binding name="Endpoint1" endpointName="Endpoint1" />
        </Bindings>
      </Site>
    </Sites>
    <Endpoints>
      <InputEndpoint name="Endpoint1" protocol="http" port="80" />
    </Endpoints>
    <Startup>
      <Task commandLine="setup.cmd" executionContext="elevated" taskType="simple">
      </Task>
    </Startup>
  </WebRole>
</ServiceDefinition>
```

最後に、ファイルの保存先を Windows Azure ストレージ サービスに変更します。upload.php の以下の部分を修正します。

```
$host = Microsoft_WindowsAzure_Storage::URL_CLOUD_BLOB;
$accountName = 'ストレージアカウント';
$accountKey  = 'ストレージキー';
$usePathStyleUri = false;
```

■Windows Azure 本番環境への展開

いよいよ Windows Azure 本番環境へ展開していきます。最初に、アップロードする本番用パッケージを作成するために、以下のコマンドを実行します。

cspack ServiceDefinition.csdef /sitePhysicalDirectories:WebRole;Web;wwwroot /role:WebRole;wwwroot

パッケージ化が終了すると、サービス構成設定ファイルと同じフォルダ内に ServiceDefinition.cspkg が作成されるので、Windows Azure デベロッパー ポータルにアクセスしてデプロイします。

Windows Azure デベロッパー ポータル
http://windows.azure.com/

デプロイが終わったら、ローカルと同様にファイルのアップロードを行い、確認してください。実際には異なる環境でアプリケーションが動作していますが、先ほどとほぼ変わりなく動作します。

ここまで、Windows Azure SDK とエディタを使った、簡単な Azure 用 PHP Web アプリケーションの開発方法と、本番環境へのデプロイについて解説してきました。
通常の IIS で稼働している既存の Web アプリケーションを Azure 上へ移行するシナリオなどでも、CSPack と CSRun を使ったデプロイ方法を適用できます。ぜひ一度触ってみてください。