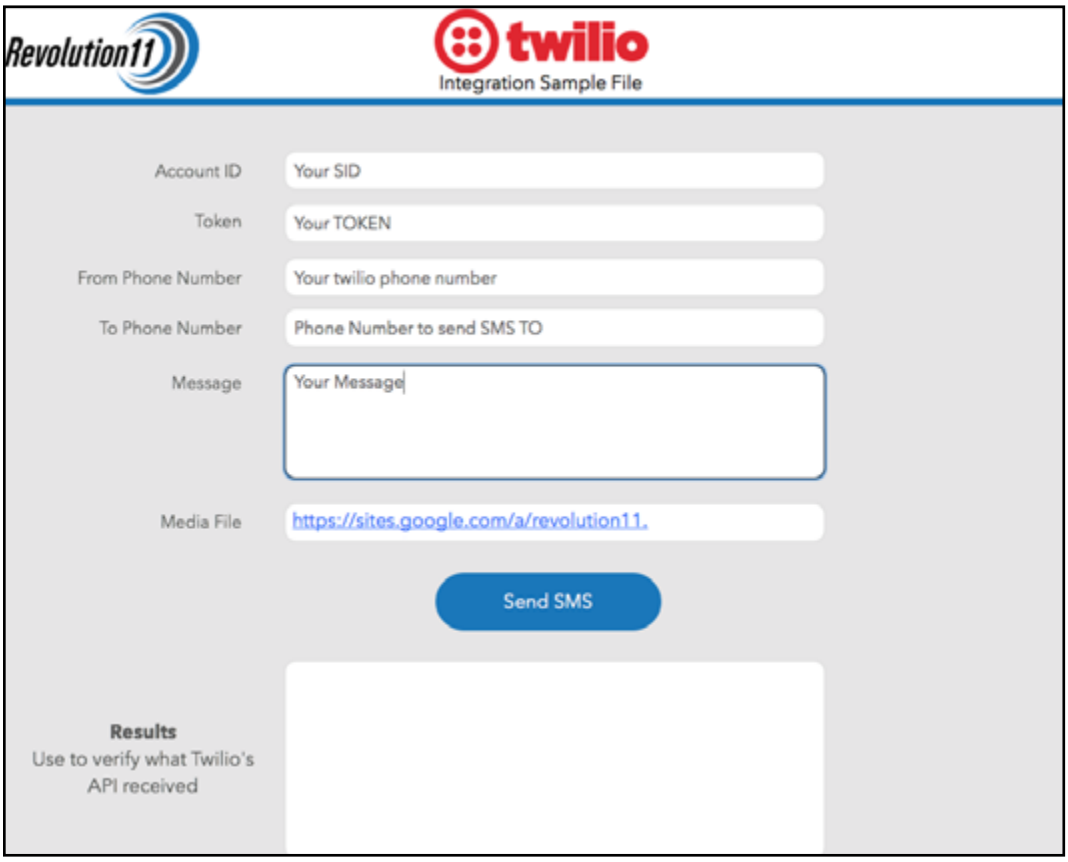
More About Twilio & FileMaker

Welcome back to Part 2 of the Revolution11 - Twilio/FileMaker integration where we have fun with Twilio, the service that allows you to make and receive phone calls and text messages using its web service APIs. In **Part 1**, we went over using a simple HTTPS request to send an SMS message, including a media file. In this post we will drill down into each individual piece that makes this integration possible and help you find the pieces to build it.

Previously we talked about:

- Integrating Twilio and FileMaker
- The PHP web service for retrieving texts from Twilio's server

Here is a screenshot of the Twilio Demo File:



As you can see from the screenshot, a few fields need to be populated:

- account_id
- token
- from_phone_number
- to_phone_number
- message

These fields are used to populate the variables used in the script to send the SMS message.

This is what the FileMaker script code looks like:

```
send_text_message
# Created by Sean on 11/29/15
# Sample file for sending SMS via Twilio

# Set variables based on fields
Set Variable [ $sms_accountID ; Value: Twilio Sample File::account_ID ]
Set Variable [ $sms_token ; Value: Twilio Sample File::token ]
Set Variable [ $to_phone_number ; Value: Twilio Sample File::to_phone_number ]
Set Variable [ $from_phone_number ; Value: Twilio Sample File::from_phone_number ]
Set Variable [ $message ; Value: Twilio Sample File::Message ]
Set Variable [ $media ; Value: Twilio Sample File::Media File ]

# Validation - Make sure there are values in the account information fields, and make sure they phone number is a correct length
If [ $sms_token = ""  or  $sms_accountID = ""  or  $from_phone_number = "" ]
    Show Custom Dialog [ "ERROR" ; "You have not entered you Twilio credentials." ]
    Exit Script [ ]
Else If [ /*Length(SMS::from_phone_number) > 10  or  Length(SMS::to_phone_number) > 10*/ ]
    Show Custom Dialog [ "ERROR" ; "The phone numbers must be 10 numbers in length. No more, No Less." ]
    Exit Script [ ]
End If

# Build the POST url to Twilio's API by putting all the pertinent information in a variable called $post
Set Variable [ $post ;
Value: "https post://" &   $sms_accountID & ":" & $sms_token & "@api.twilio.com/2010-04-01/Accounts/" & $sms_accountID & "/Message… ]

# Show results from API Call in the Results Field. Checking the results can validate if it was sent successfully
Insert from URL [ Select ; No dialog ; Twilio Sample File::results ; $post ]
Halt Script
```

The script checks that the required fields are not empty and that the phone numbers are formatted properly. The $post variable is set using the variables named earlier in the script, which sends a request to Twilio's API. Twilio's API takes the data that you are passing through this transaction, validates your Twilio account, and sends the message out.

Pretty fancy! But what happens when someone replies to that SMS? We can handle the reply easily with just a touch of PHP. In our example we are sending the replies to the SMS straight to an email address.

```php
<?php
// PHP file hosted on a local file
require "C:YOURFILEPATH/twilio.php";

/**
 * This allows you to output a twilio response
 */

header('Content-type: text/xml');
echo '<?xml version="1.0" encoding="UTF-8"?>';


echo '<Response>
        <Sms>thanks for the support ticket, we will get back to you shortly</Sms>
</Response>'; //Place the desired response here


/**
 * This section actually sends the email to support.
 */

$to      = "email1@yourdomain.com, email2@yourdomain.com, email2@yourdomain.com"; // Your email address

$subject = "Message from {$_REQUEST['From']} at {$_REQUEST['To']}"; //Number this is

$message = "You have received a message from {$_REQUEST['From']}.

Body: {$_REQUEST['Body']}";

$headers = "From: myEmail@yourdomain.com"; // Who should it come from?


mail($to, $subject, $message, $headers);
```

This PHP file is kept on your FileMaker server, with the standard web configuration turned on. In the example, we have used “Your file path” so you can configure the PHP file for your server.

The header and echo tags are basically telling our application that this is the type of data that we are going to be receiving from Twilio’s server.

The next portion is variable creation and assignment. For example:

- The $to variable - where we are sending the response message once it’s retrieved from Twilio’s server
- The $subject variable - The subject of the message
- The $message - The actual message and,
- The $headers variable - the person the email was sent from

The final section is an HTTP request method that will send the reply email.

You obviously don’t want to have to fool with the PHP files on your server every time you want to change the reply action to the outgoing SMS message, so stay tuned for a future Revolution11 blog post where we show you a few ways to get these preferences out of your FileMaker solution.