



# Integrating SMS into Library Services @ Randall Library

Jason Fleming, Garret Corbett



Scan for a link to a copy of the poster

## Abstract

Randall Library at the University of North Carolina Wilmington (UNCW) negotiated the purchase of Twilio SMS service. Using their development toolkit, we have integrated several numbers into various library services.

## Purchasing Twilio

- With a number of new SMS based initiatives being considered by the library we felt it best to purchase a phone number that we could send and receive texts from. With Twilio we were able to secure a line and add credit for continued use at an affordable rate
- Several meetings with the library business officer and purchasing got us to the point where it could move forward and we could start actively trying to purchase the service
- The main hurdle was the fact that it wasn't a one time purchase and it also wasn't really a subscription. So there wasn't a lot of precedent for this spending money this way

## Library Catalog

- In the interest of helping students find books they are looking for we provide them a way to send themselves a call # for a book they want to retrieve
- For some time now we have been using the bruteforce javascript method of sending an SMS that required the user to indicate what cell phone provider they used
- Once we had access to the Twilio API we were able to update the service by integrating it into our ILS using Javascript to provide this stylish interface:

Send Item Information To:

## Integrating Twilio using Javascript

- First install twilio using the node package manager:

***npm install twilio***

- Second, you simply need to ***require('twilio')*** and send the message:

```
router.post('/', function (req, res) {
  //Connect the Twilio Client
  var client = require('twilio')(process.env.TWILIOSID,
                                process.env.TWILIO_TOKEN);

  //Create and send the text Message
  client.messages.create({
    to: req.body.phone,
    from: process.env.TWILLIONUMBER,
    body: req.body.message,
  }, function(err, message) {
    console.log(message.sid);
  });
  res.redirect('/');
});
```

## ILLIAD

We can now notify ILL patrons about their requests, users can text us questions that come in to our chat service, and users can text themselves call #'s from our catalog. This was accomplished by installing the Illiad addon and putting in the Twilio credentials.

<https://prometheus.atlas-sys.com/display/ILLiadAddons/ILLiad+Twilio+SMS+Addon>

## Chat Reference

- We use Libraryh3lp as our chat reference service and their application makes it easy to setup SMS interactions using Twilio.
- As part of their (libraryh3lp) service they offer a way to purchase Twilio through them, but as stated previously we purchased it ourselves
- Try it yourself by Texting us @ (910) 218-0782

## Escape Room Reservations

Over the Fall of 2017, UNCW ran an escape room, and as part of the reservation system we enabled an option to receive text messages that was heavily used

## Integration Using PHP in a Laravel Application:

- First install Twilio using the composer:

***composer require twilio/sdk***

- Second, you simply need to ***use Twilio\Rest\Client;*** in the file that you would like to send the message

- Third, you need to construct the message to send

```
function __construct()
{
    $this->appointments = \App\Appointment::appointmentsDue()->get();

    $twilioConfig = config('services.twilio');
    $accountSid = $twilioConfig['twilio_account_sid'];
    $authToken = $twilioConfig['twilio_auth_token'];
    $this->sendingNumber = $twilioConfig['twilio_number'];

    $this->twilioClient = new Client($accountSid, $authToken);
}
```

\*For this application, we also set up a job that ran every minute to see if any texts needed to be sent based on a form where users could have signed up for text notifications. This was done in the Kernel.php file using the following code:

```
<?php
namespace App\Console;
use Illuminate\Console\Scheduling\Schedule;
use Illuminate\Foundation\Console\Kernel as ConsoleKernel;
class Kernel extends ConsoleKernel
{
    protected $commands = ['\App\Console\Commands\SendReminders'];
    protected function schedule(Schedule $schedule)
    {
        $schedule->command('reminders:send')->everyMinute();
    }
}
```

## Ideas for the Future

The next project we are working on is to send users notifications using text messages from our Integrated Library System.