We left off with a nearly functional email scene. We could use iOS' built in email composing components (search the documentation for MFMailComposeViewController). There are a few shortcomings of the native mail experience, though:

- The user may not have the built in Mail app configured (and thus be unable to send a message).
- The user may not want to expose their email address to you.

A better experience would be to manage the mailing process for them, and not require an email address. We can accomplish this relatively easily via Sendgrid (or any other 3rd party emailing solution). Plus, it's a great intro to dealing with third party libraries.

Introducing Cocoapods

Cocoapods is the defacto standard for managing third party libraries within an iOS application. It is both a database of available libraries, as well as a system that manages installing and updating them for you.

Search around the Cocoapods site to get a sense for the vast array of third party code that is available (try searching for menu libraries, animation tools, etc).

However, getting started will require diving into the terminal a bit (Cocoapods is a very developer-focused tool).

Installing Cocoapods

The first step is to install Cocoapods. Open up the Terminal app (you can find it under Applications/Utilities). You will want to run the following commands:

```
hostname:~ user$ export PATH=/Applications/Xcode.app/Contents/Developer/usr/bin:$PATH
hostname:~ user$ sudo gem install cocoapods
Password:
```

If you're new to the terminal:

- Press return after each line to execute it.
- When typing your password (your computer password), you will not see anything on screen! It's a security feature :)
- After you type your password correctly, nothing should appear to happen for a while. That's good! It's downloading and installing Cocoapods.

After a bit of a wait, you should see a bunch of text fly by, and end up with something along the lines of:

```
Successfully installed cocoapods-0.29.0
1 gem installed
hostname:~ user$
```

You may also get a prompt asking whether or not you wish to install over rake. Say yes.

The Podfile

The last thing you need to do is to write a Podfile for the project. A Cocoapods Podfile describes all of the libraries and their versions that your project depends on.

You need to change to the directory that your Xcode project resides within. Type cd followed by the path to your project (if you're unsure - type cd and then drag the project's folder into the terminal - it will paste the path):

```
hostname:~ user$ cd /Users/user/The\ Path\ To/Your/Project\ Directory/Resume
```

From that directory, we want to create a Podfile:

```
hostname:Resume user$ echo "platform :ios, '7.0'
pod 'sendgrid', '~> 0.1.0'" > Podfile
```

Finally, you can instruct Cocoapods to install the send grid library:

```
hostname:Resume user$ pod install
Analyzing dependencies
Downloading dependencies
Installing AFNetworking (2.1.0)
Installing sendgrid (0.1.0)
Generating Pods project
Integrating client project

[!] From now on use `Resume.xcworkspace`.
[deprecated] I18n.enforce_available_locales will default to true in the future. If you really
 want to skip validation of your locale you can set I18n.enforce_available_locales = false to
 avoid this message.
hostname:Resume user$
```

Do as it instructs: Close Xcode entirely, and then reopen the **xcworkspace** file in your project directory. *Never* open the xcodeproj file again!

You should now have a project integrated with Cocoapods and the Sendgrid library!

If you are having trouble getting Cocoapods installed work working, feel free to check out or download the 14-dependencies branch of the project.

Sending Email

Home stretch! In order to integrate our email view with Sendgrid, we need to let our code know about it. You can find documentation for sendgrid's iOS library on their GitHub repository: https://github.com/sendgrid/sendgrid-objc.

Open **EmailViewController.m** and add the following import line to the top:

```
#import "EmailViewController.h"
#import <sendgrid.h>
```

All we have to do next is fill in the send: action:

```
- (IBAction)send:(id)sender
{
    sendgrid *message = [sendgrid user:@"username" andPass:@"password"];

    message.to = @"your@email.com";
    message.from = @"your@email.com";
    message.subject = [NSString stringWithFormat:@"Resume App Message From: %@\n\n",
                       self.senderTextField.text];
    message.text = self.bodyTextView.text;
    message.html = self.bodyTextView.text;

    [message sendWithWeb];
    [self.navigationController popViewControllerAnimated:YES];
}
```

Don't forget to replace username and password with your sendgrid username and password. Also, the to and from email address should point to an actual email address of yours!

Recall that we didn't want the user to enter their own email address, so we're just using our own for the from.

Run it in the simulator, type in a message and hit send! If all is well, you should see it show up in your inbox within a minute or two.

Great success!