Guide: Custom Salesforce Outbound SMS Button

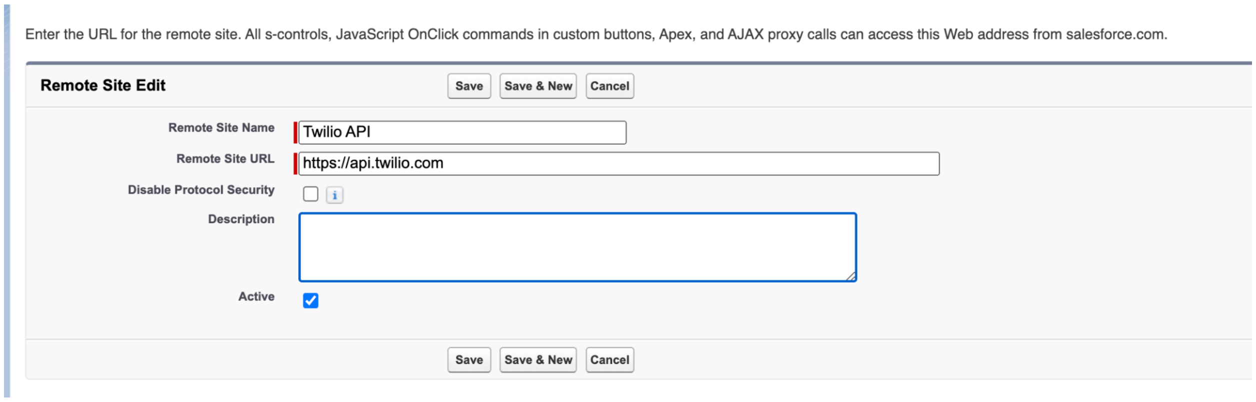
Prerequisites

1. Have a Twilio Account
2. Setup an API Key for the integration
3. Have a DID available
   a. Required to have A2P DLC registered or outbound will fail
4. Have a Salesforce Account
5. Grab the code from this repo

Configuration

Salesforce Steps

1. Add Remote Site
   a. Login to Salesforce > Open Setup > Remote Site Settings: and New Remote Site
      i. Name = TwilioAPI
      ii. Site = https://api.twilio.com

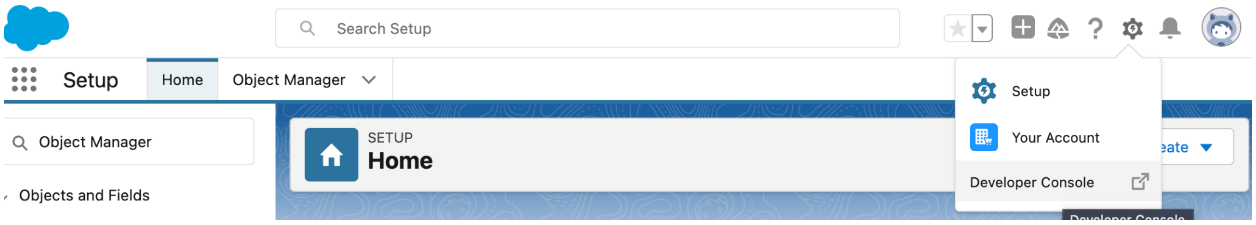


Enter the URL for the remote site. All s-controls, JavaScript OnClick commands in custom buttons, Apex, and AJAX proxy calls can access this Web address from salesforce.com.

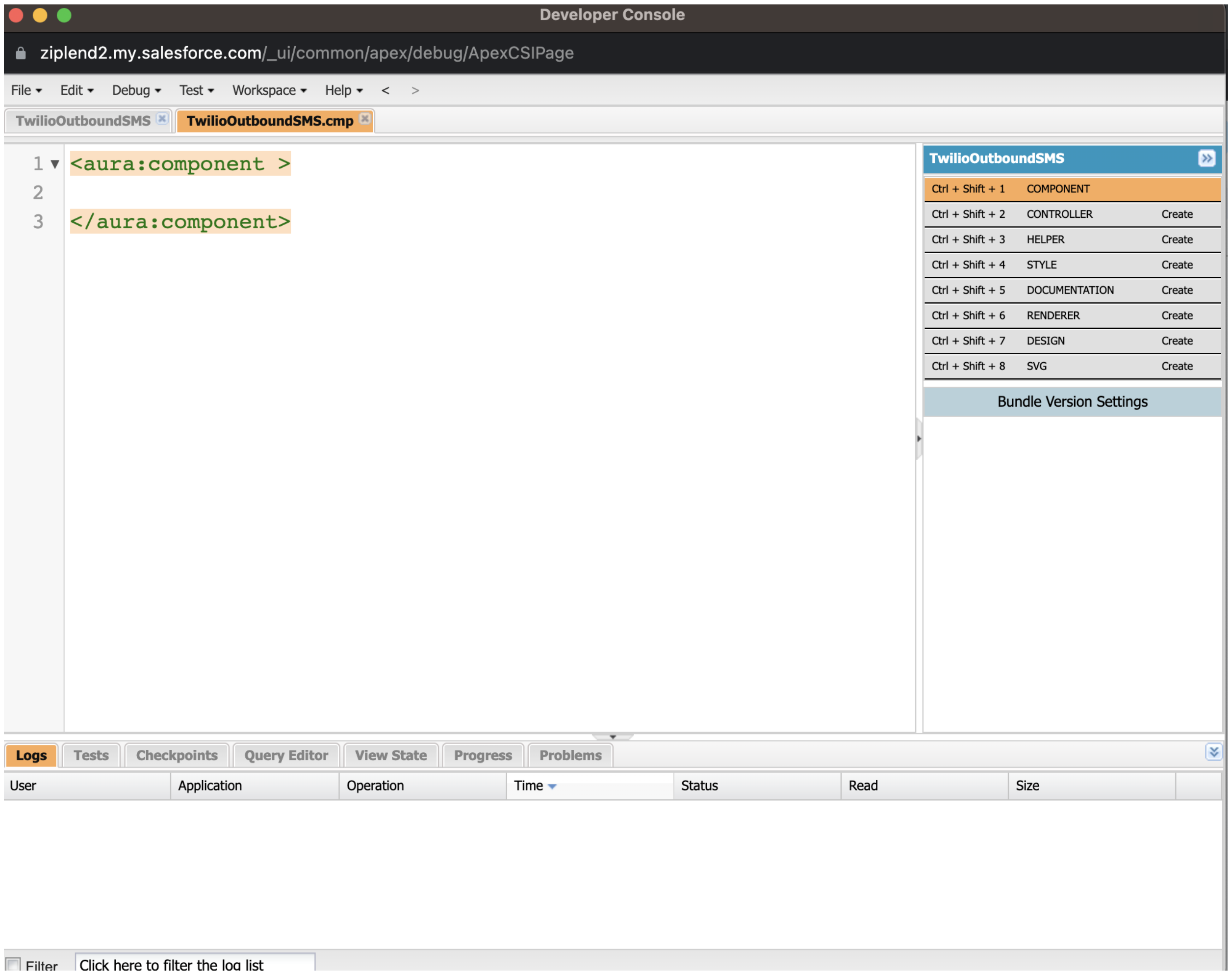
Remote Site Edit

Remote Site Name	Twilio API
Remote Site URL	https://api.twilio.com
Disable Protocol Security	☐
Description	
Active	✓

2. Import Apex Classes
   a. Login to Salesforce > Open Setup > Apex Classes: and Click New
   b. We will be create two Apex Classes (copy the code and save for each of the below)
      i. twilioSmsApexClass
         1. **Update the Account SID, API Key, API Secret, and From number before saving**
      ii. Contacts
   c. Special Note - If you are attempting to deploy this into a Production Salesforce environment you won't have the option to, see Appendix below if you are in production. If you are on a Test/Dev/Sandbox you will have no issues :)
3. Create the Lightning Component
   a. Login to Salesforce > Click the Settings Cogwheel > Select Developer Console



b. In the Developer Console > Click File > New > Lightning Component
c. Give it a name and description (remember this later)
    i. Example Name: TwilioOutboundSMS
    ii. Example Description: Component used to send Outbound SMS leveraging Twilio



```
<aura:component >

</aura:component>
```

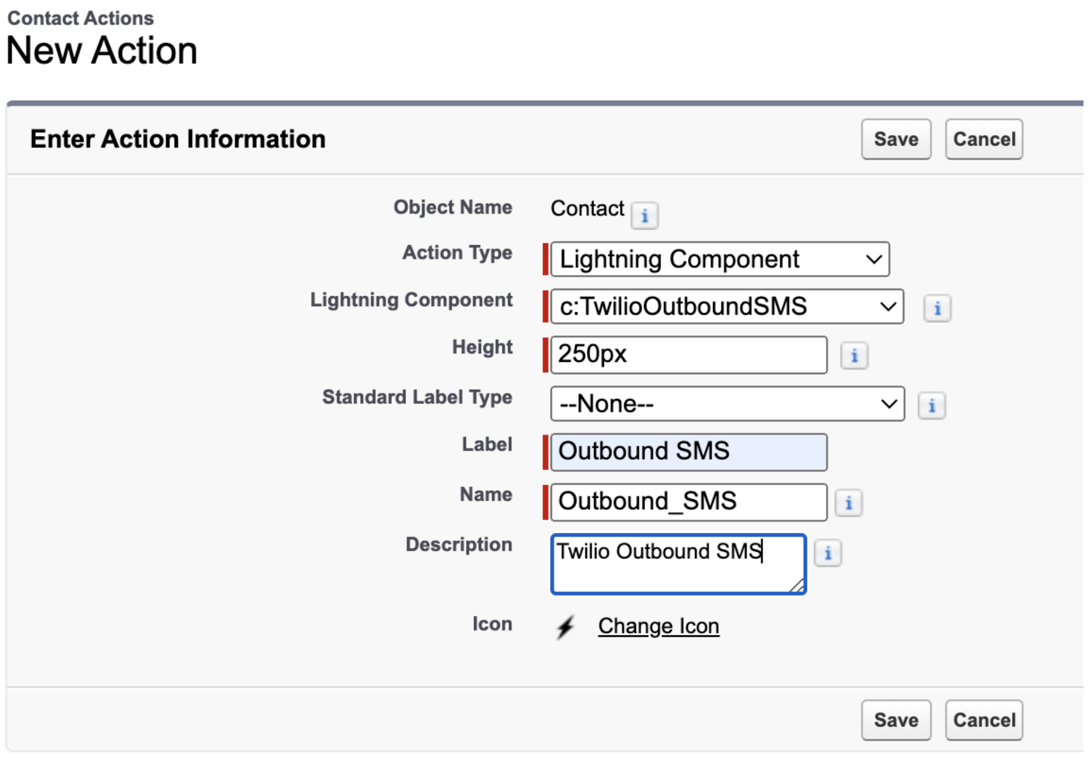
Ctrl + Shift + 1	COMPONENT	
Ctrl + Shift + 2	CONTROLLER	Create
Ctrl + Shift + 3	HELPER	Create
Ctrl + Shift + 4	STYLE	Create
Ctrl + Shift + 5	DOCUMENTATION	Create
Ctrl + Shift + 6	RENDERER	Create
Ctrl + Shift + 7	DESIGN	Create
Ctrl + Shift + 8	SVG	Create

d. Copy the Code into the respective sections on the right side and save each of them:
    i. Controller
    ii. Helper
    iii. Component
    iv. Design
    v. Style

4. Configure a Quick Action to the Contacts Page:

**Special note* for this example we are building this action on the Contacts Page, it can be added to any page following similar steps just select another page in the Object Manager*

a. Create the Button: Login to Salesforce > Open Setup > Navigate to Object Manager > Contact > Buttons, Links, and Actions: Click the New Action (top right of the page)
   i. Action Type = Lightning Component
   ii. Lightning Component = (What you named it above - Example TwilioOutboundSMS
   iii. Label = (This is what the button label will be) Example Outbound SMS
   iv. Name = this auto populates
   v. Click Save



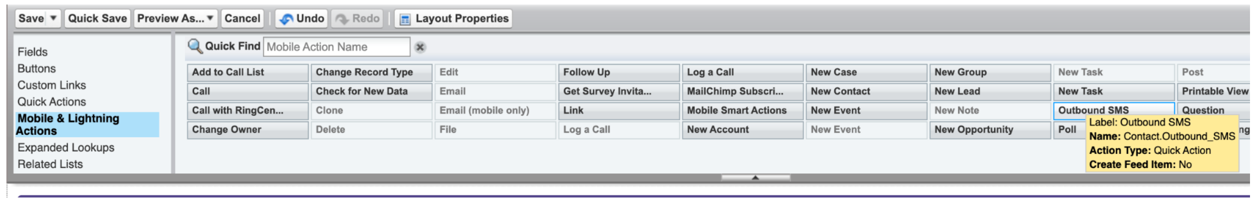
Contact Actions

New Action

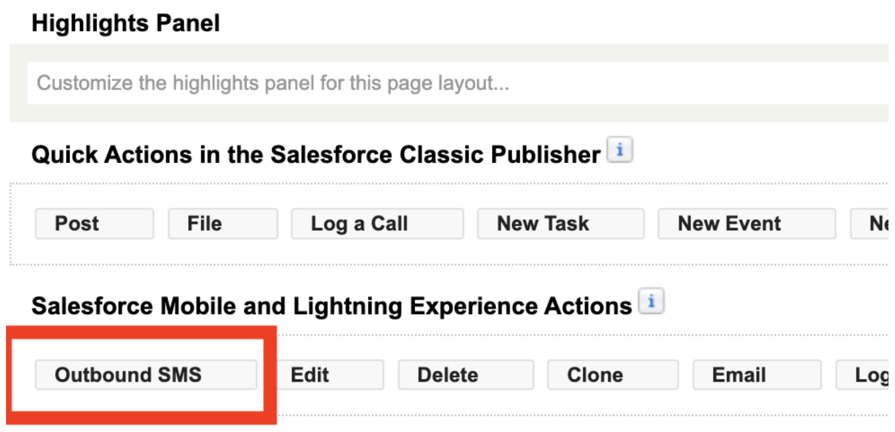
Enter Action Information

Object Name	Contact
Action Type	Lightning Component
Lightning Component	c:TwilioOutboundSMS
Height	250px
Standard Label Type	--None--
Label	Outbound SMS
Name	Outbound_SMS
Description	Twilio Outbound SMS
Icon	Change Icon

b. Change the Page Layout: Setup > Object Manager > Contact > Page Layouts > Contact Layout
   i. Select Mobile & Lightning Actions > You should see the above Action you just created. Drag it into the Salesforce Mobile and Lightning Experience Actions section (Put it towards the left)
   ii. Click Save

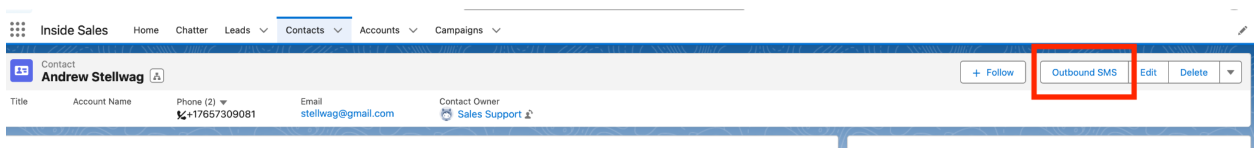


Label: Outbound SMS
Name: Contact.Outbound_SMS
Action Type: Quick Action
Create Feed Item: No


Highlights Panel
Customize the highlights panel for this page layout...
Quick Actions in the Salesforce Classic Publisher
Post
File
Log a Call
New Task
New Event
Salesforce Mobile and Lightning Experience Actions
Outbound SMS
Edit
Delete
Clone
Email

Testing

1. If all things went well, if you navigate to a normal Salesforce Section > Contacts tab, you should see the Outbound SMS button!



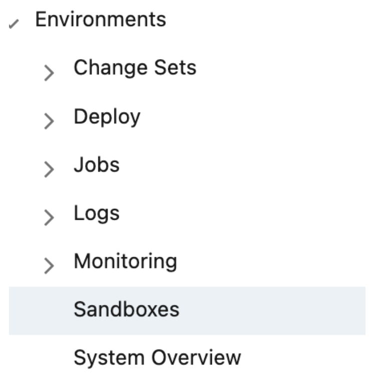
   a. Where it's located on the page may differ based on the page layout for each customer/org
2. To test, click Outbound SMS, it will send a sms to the phone number for that contact
3. By default it has some canned responses, this can be easily modify by changing the Component.html code that you imported via the Developer Console
4. You are able to also text in the textbox free form and send, you can also click the canned response and alter it before sending
5. Have fun!

Appendix

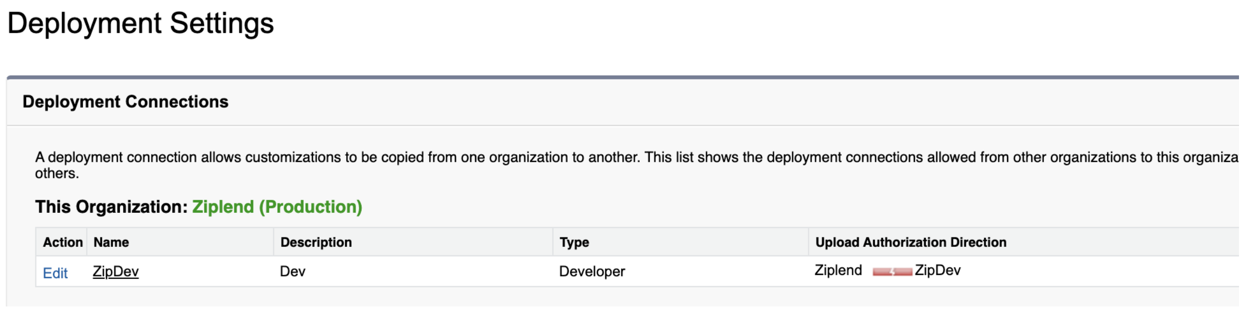
Production Salesforce Apex Class Deployment Special Steps

1. Creating & Deploy Apex Classes into a Production Salesforce Environment:
   a. NOTE this is very easy to deploy in development org. If deploying in a production org it requires you to have a sandbox connected to the production salesforce environment and you are able to deploy code from sandbox to production. It will not allow you to create an Apex class in production

i. If you are in a Production org create a Sandbox via Salesforce > Setup > Search for Sandbox and create one from there



1. It'll take a few minutes to create, but you should be able to login from there and complete the Apex Class Steps from there and then deploy them to your Production environment
    a. Logging into a sandbox - https://help.salesforce.com/s/articleView?id=sf.data_sandbox_create.htm&type=5
2. You'll only be required to do this for Apex Classes, everything else can be completed in the Production Environment
3. If you have the Sandbox setup properly, in your Production Org under Setup > Deployment Settings, you should see something like this:



Deployment Settings

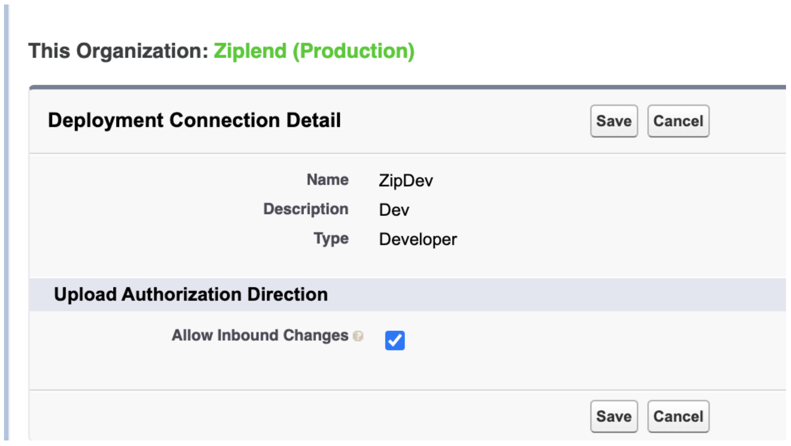
Deployment Connections

A deployment connection allows customizations to be copied from one organization to another. This list shows the deployment connections allowed from other organizations to this organiza
others.

This Organization: Ziplend (Production)

Action	Name	Description	Type	Upload Authorization Direction
Edit	ZipDev	Dev	Developer	Ziplend ⇄ ZipDev

4. Make sure once it is done copying to Edit that it is allowed to do Inbound Changes


This Organization: Ziplend (Production)

Deployment Connection Detail

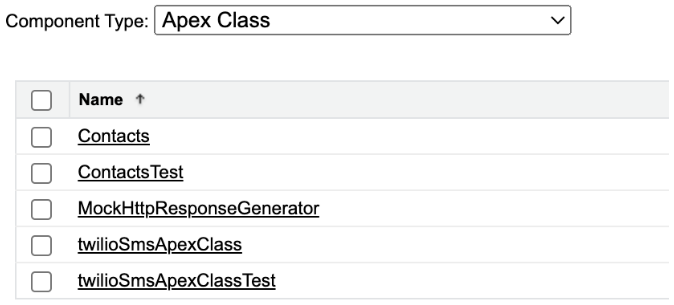
Name	ZipDev
Description	Dev
Type	Developer

Upload Authorization Direction

Allow Inbound Changes ✓

ii. We will need to Deploy a few extra Apex Classes in your Dev/Sandbox, these will be required due to Salesforce requiring any Production Apex Classes to have Unit Tests. In the repo, you'll find the below ApexClass, create those Apex Classes (IE Setup > Apex Classes > New)
   1. Add twilioSmsApexClassTest-1 + twilioSmsApexClassTest-2 + Contacts Test

iii. Deploy the above Apex Classes from the Sandbox > Production: https://help.salesforce.com/s/articleView?id=000382677&type=1
   1. In you Dev/Sandbox Salesforce go to Setup > Outbound Change Sets > click New > Create a name/description > Click Save
      a. You'll need to do this for the Contacts and twilioSmsApexClass Apex Classes
      b. Under Change Set Components > click Add > for the Component Type Select Apex Class (You should see all the Apex Classes you previously created)


Component Type: Apex Class

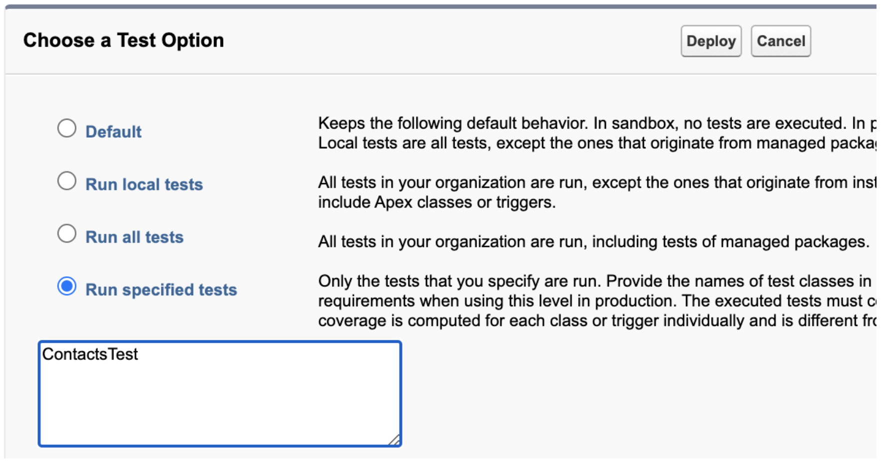
	Name ↑
☐	Contacts
☐	ContactsTest
☐	MockHttpResponseGenerator
☐	twilioSmsApexClass
☐	twilioSmsApexClassTest

      c. We will end up creating two different Outbound Change Sets (one for Contacts and one for twilioSmsApexClass)
      d. For Contacts: Select Contacts + Contacts Test, then click Add Change Set > in the next screen click Deploy and it will have you select the Production org
      e. Create another for twilioSmsApexClass: Select MockHttpResponseGenerator + twilioSmsApexClass + twilioSmsApexClassTest, then click Add Change Set. in the

next screen click Deploy and it will have you select the Production org

iv. Once you've completed that in the Production instances go to Setup > Inbound Change Sets, you should see the code you deployed from the sandbox, you'll need to click "Deploy" for the Apex Classes to be deployed

1. Contacts + ContactsTest
    a. When you deploy > Select Specific Tests and type "ContactsTest" - This will allow the Apex Class to pass the tests and deploy successfully



Choose a Test Option

Default	Keeps the following default behavior. In sandbox, no tests are executed. In p Local tests are all tests, except the ones that originate from managed packa
Run local tests	All tests in your organization are run, except the ones that originate from inst include Apex classes or triggers.
Run all tests	All tests in your organization are run, including tests of managed packages.
[✓ Run specified tests]	Only the tests that you specify are run. Provide the names of test classes in requirements when using this level in production. The executed tests must c coverage is computed for each class or trigger individually and is different fr

ContactsTest

2. twilioSmsApexClass + twilioSmsApexClassTest-1 + twilioSmsApexClassTest-2
    a. When you deploy > Select Specific Tests and type "twilioSmsApexClassTest" - This will allow the Apex Class to pass the tests and deploy successfully