**Calling Microsoft Flow from your application**

**Overview**

Microsoft Flow allows you to turn complex repetitive tasks into automated multistep flows. In many cases, you want your flow to be triggered by an event, such as a document update. In others, you want to let the end user initiate a flow. In this case, Microsoft Flow will serve as an integration glue between your application and large number of other services. For example, you may want the user to click on a button and send an email to her customers.

**Functionality**

When first adding the ‘When a HTTP request is received’ trigger, to a flow you’re presented with a HTTP POST URL informing you that the URL will be generated after the Flow has been saved. This means that while you’re initially creating your Flow, you will not be able to provide/use the URL to that is required to trigger the Flow.

**Step 1**

Before we start implementation, let's think about the format of the call. In this example, triggering an email from application, we want to pass the following information from the client to the flow: email address, email subject and the text of the email. Our request body will look like this:

{

"emailaddress":"jenkinsns@jpower4mvp.onmicrosoft.com",

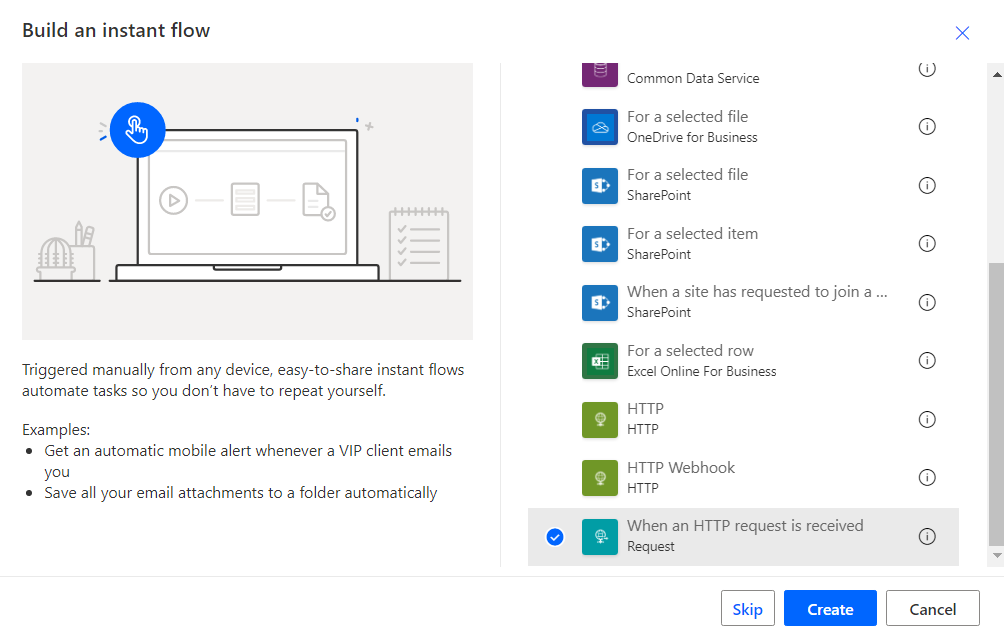
"emailSubject": "HTTP Subject",

"emailBody": "Welcome to Calling Microsoft Flow from my custom application"

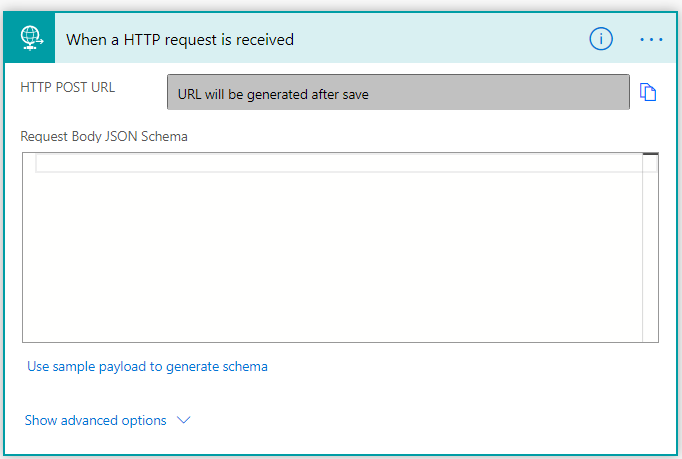
}

**Step 2**

1. Create a Instant flow 🡪 Select ‘**When an HTTP request is received’**



1. Click Create to Create flow



1. When first adding the ‘When a HTTP request is received’ trigger, to a flow you’re presented with a HTTP POST URL informing you that the URL will be generated after the Flow has been saved.
2. Click ‘Use sample payload to generate schema’
3. Then Copy/paste below code to generate Request Body JSON Schema

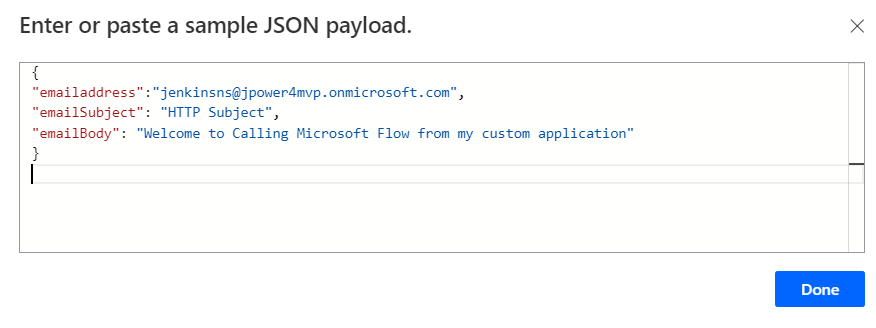
*{*

*"emailaddress":"jenkinsns@jpower4mvp.onmicrosoft.com",*

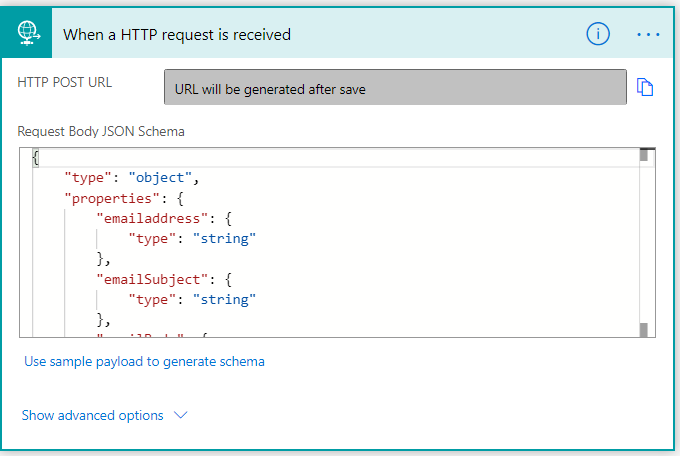
*"emailSubject": "HTTP Subject",*

*"emailBody": "Welcome to Calling Microsoft Flow from my custom application"*

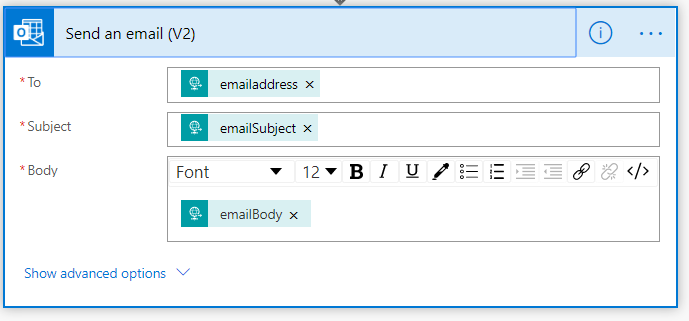
*}*



1. Click done to generate Request Body JSON Schema, refer below screenshot to get the schema



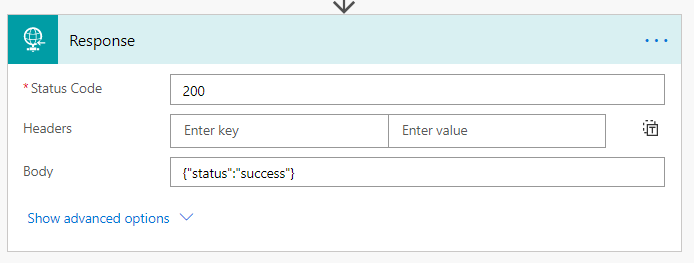
1. Add an action to send email



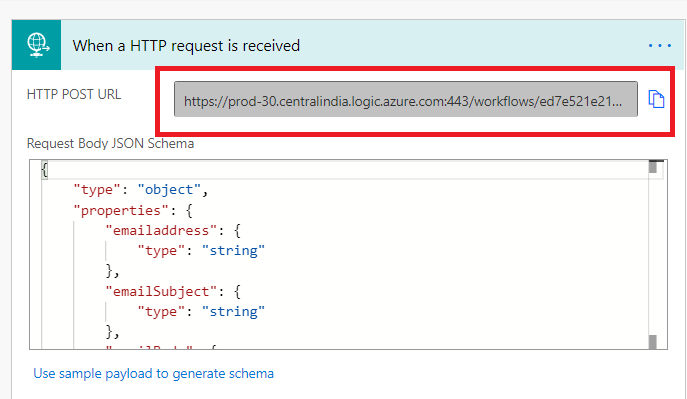
1. Select the inputs from HTTP request
   1. Email address
   2. Email Subject
   3. Email Body
2. Then save the flow

**Step 3**

1. Click again on the "+New step" and choose the "Response" action
2. Here you have an option to either specify the JSON body of the response, such as {"status":"success"} or use the Body of the original request by clicking on the "Body" property.



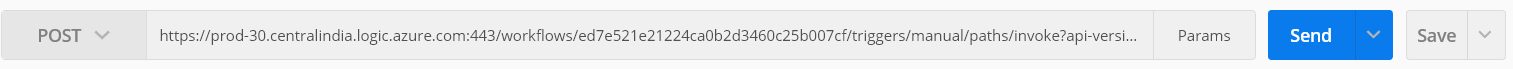
1. Go to trigger ‘When a HTTP request is received’ and copy the url



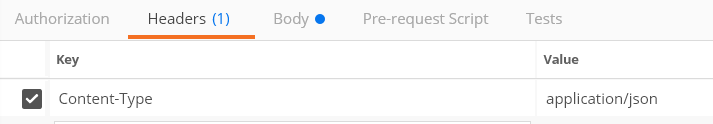
**Step 4**

**Test your flow**

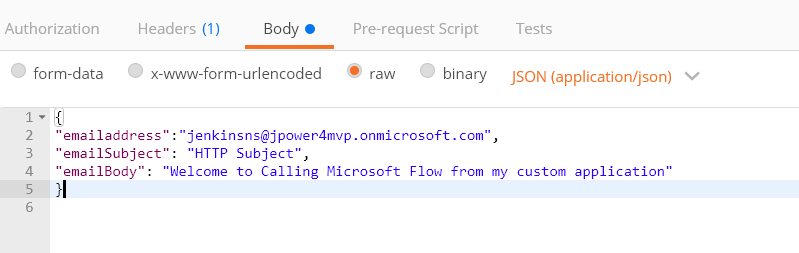
1. Go to Postman chrome extension
2. Add the copied URL



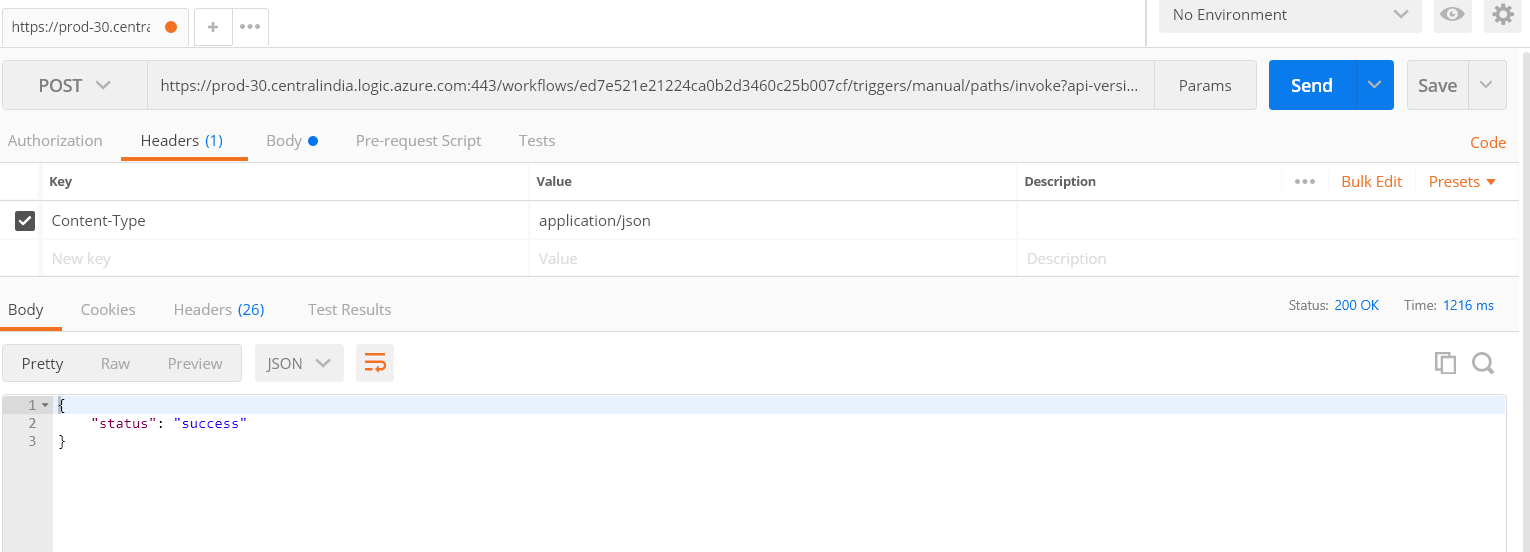
1. Add header



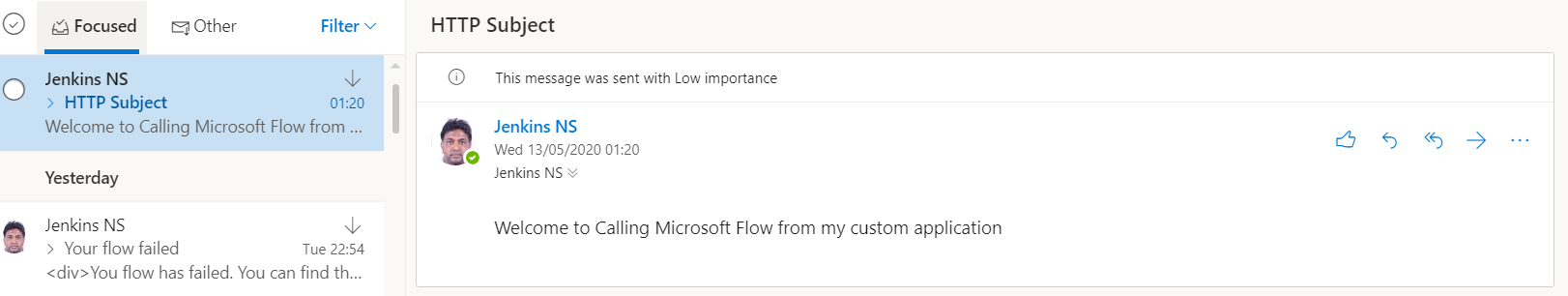
1. Add body request as raw



1. Click Send to execute the event
2. It will reply the status as Success



1. Go to email and check the email



Test from SPFx

Create a new webpart

Text

Description automatically generated

Add below code in the webpart class- > replace render method

public render(): void {

    this.domElement.innerHTML = `

    <section class="${styles.spfxPawpExample} ${!!this.context.sdks.microsoftTeams ? styles.teams : ''}">

      <div class="${styles.welcome}">

        <h3>Welcome to SharePoint Framework!</h3>

        <p>

        <button type="button" id="btn">Send Email</button>

        </p>

      </div>

    </section>`;

    this.\_setbtnEventHandlers();

  }

  private \_setbtnEventHandlers():void{

    this.domElement.querySelector('#btn').addEventListener('click',()=>{

      this.sentEmail();

     } );

  }

  private sentEmail():Promise<HttpClientResponse> {

    const posturl = "https://prod-16.centralindia.logic.azure.com:443/workflows/7da8d94469ac44449d3d74e4c2fd4145/triggers/manual/paths/invoke?api-version=2016-06-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=uzN7yjlpd1M1m60-yWt6kVMdr3B87U7UBg6FNHDX9yQ";

    const bodystr:string = JSON.stringify({

      'Email': "admin@jenkinsnsfs.onmicrosoft.com",

      'Subject': "Welcome to SPFx event to Power Automate",

      'Body': "This email from SPFx"

    });

    const requestheaders:Headers = new Headers();

    requestheaders.append('Content-type','application/json');

    const httpsoptions:IHttpClientOptions = {

      body:bodystr,

      headers:requestheaders

    };

    return this.context.httpClient.post(

      posturl,

      HttpClient.configurations.v1,httpsoptions).then((response:HttpClientResponse):Promise<HttpClientResponse> =>{

        return response.json();

      });

  }

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated