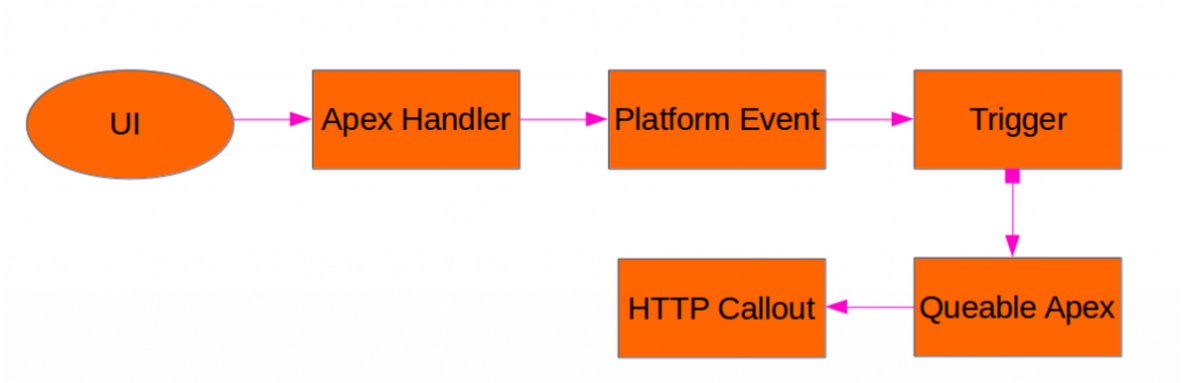
Using Platform Events with Queueable Apex

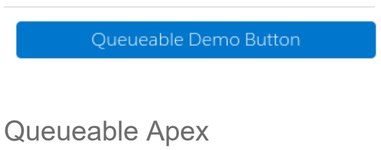
By ufthelp March 4, 2020

No Comments



Scenario:- How to call Queueable Apex inside platform events

Step1:- Have a button(can be vlocity action or vlocity Omniscript guided flow) on SFDC UI to call the “Apex Handler”



Step2:- “Apex Handler” will call Platform Event

```
 //calling PE with the required params
Demo_Platform_Event__e peObj = new Demo_Platform_Event__e( EndPoint_Url__c = endPointUrl );
Database.SaveResult sr = EventBus.publish(peObj);
if (sr.isSuccess()) {
    System.debug('Successfully published Demo_Platform_Event__e Event.');
} else {
    for(Database.Error err : sr.getErrors()) {
        System.debug('Error returned on Publishing Demo_Platform_Event__e Event: ' +
                     err.getStatusCode() +
                     ' - ' +
                     err.getMessage());
    }

}
```

Step3:- “Platform Event” when called will launch the “Trigger”

Setup > Platform Events > New Platform Events

Add Trigger and Custom Fields as shown below:-

Platform Event

Demo Platform Event

Standard Fields [3] | Custom Fields & Relationships [1]

Platform Event Definition Detail Edit Delete

Singular Label	Demo Platform Event	Description	
Plural Label	Demo Platform Events	Deployment Status	Deployed
Object Name	Demo_Platform_Event		
API Name	Demo_Platform_Event__e		
Event Type	High Volume		
Publish Behavior	Publish Immediately		
Created By	uftHelp, 03/03/2020 5:12 PM	Modified By	uftHelp, 03/03/2020 5:21 PM

Standard Fields

Action	Field Label	Field Name	Data Type	Controlling Field	Indexed
	Created By	CreatedBy	Lookup(User)		
	Created Date	CreatedDate	Date/Time		
	Replay ID	ReplayId	External Lookup		

Custom Fields & Relationships New

Action	Field Label	API Name	Data Type	Indexed	Controlling Field	Modified By
Edit | Del	EndPoint Url	EndPoint_Url__c	Text(100)			uftHelp, 03/03/2020 5:12 PM

Triggers New

Action	Name	Api Version	Status	Size Without Comments	Last Modified By
Edit | Del	demoTriggerHandler	48.0	Active	77	uftHelp, 03/03/2020 5:21 PM

Subscriptions

Action	Subscriber	Latest Processed Id	Latest Published Id	State
	demoTriggerHandler	1139031	-1	Running

PlatformEvent

Step4:- Inside “Trigger” call the Queueable Apex

Here we can bulkification triggerto a batch size of 50, as supported by Queueable.

```
trigger demoTriggerHandler on Demo_Platform_Event__e(after insert) {
    List<String> peListItems = new List<String>();
    Integer  counter = 0;
        for(Demo_Platform_Event_e__e peObj : peListItems) {
           counter++;
           System.debug('Counter = '+ counter);
           if (counter >=  50) { //taking batch size of 50 Id's as supported by Queueable Apex
               break;
           }
            else{
                peListItems.add(peObj);
           }
           //platform event bus
           EventBus.TriggerContext.currentContext().setResumeCheckpoint(peObj.ReplayId);
       }
       // instantiate a new instance of the Queueable class and pass params
       calloutHelper objCallout = new calloutHelper(peList);
       ID jobID = System.enqueueJob(objCallout);
       System.debug('jobID = '+ jobID);
}
```

Step5:- “Queueable Apex” will initiate “HTTP callout” to the external system.

```
      //Sample HTTP callout
      Http http = new Http();
      HttpRequest req = new HttpRequest();
      req.setEndpoint(endPointUrl);
      req.setMethod('PUT');
      req.setBody(inputData);

      req.setHeader('content-type', 'application/json');
      req.setHeader('Accept', 'application/json');
      HTTPResponse res = http.send(req);
      outputMap.put(KEY_RESULTS, res.getBody());
      Integer statusCode = res.getStatusCode();
      System.debug('code ='+statusCode);
      if(statusCode == 200){ //do logic on the success }
      else{ //throw exception }
```

More on Queueable Apexcode base