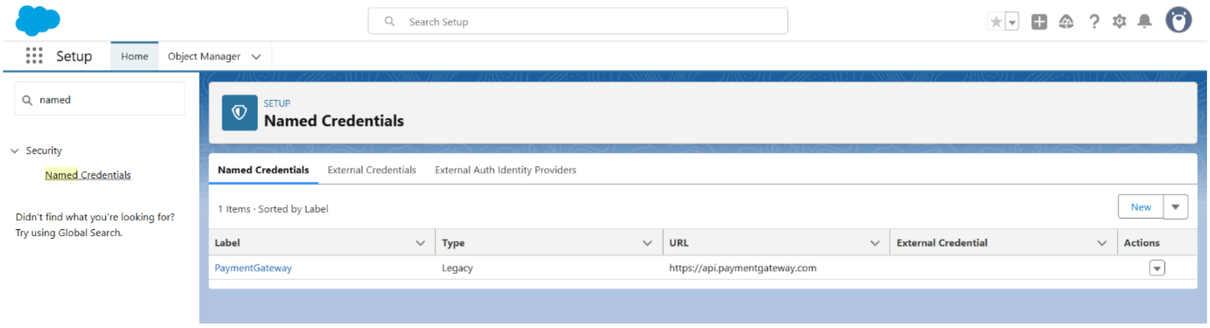
Phase 7: Integration & External Access

❖ Named Credentials:

- Created a Named Credential to define the external payment API endpoint.
- This sets up a secure connection for callouts, even if authentication is not required.

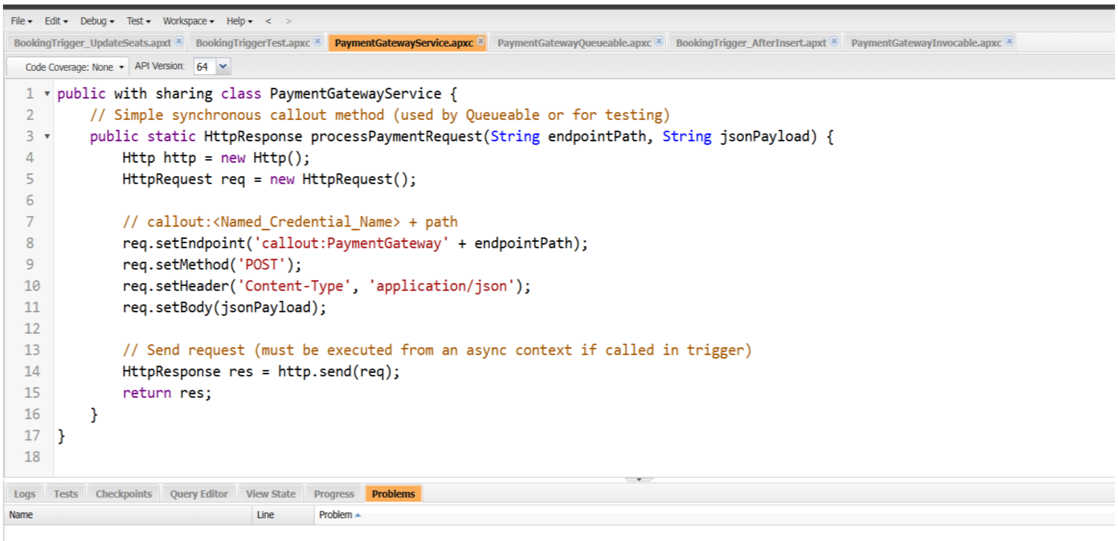


1 Items · Sorted by Label

Label	Type	URL	External Credential	Actions
PaymentGateway	Legacy	https://api.paymentgateway.com		

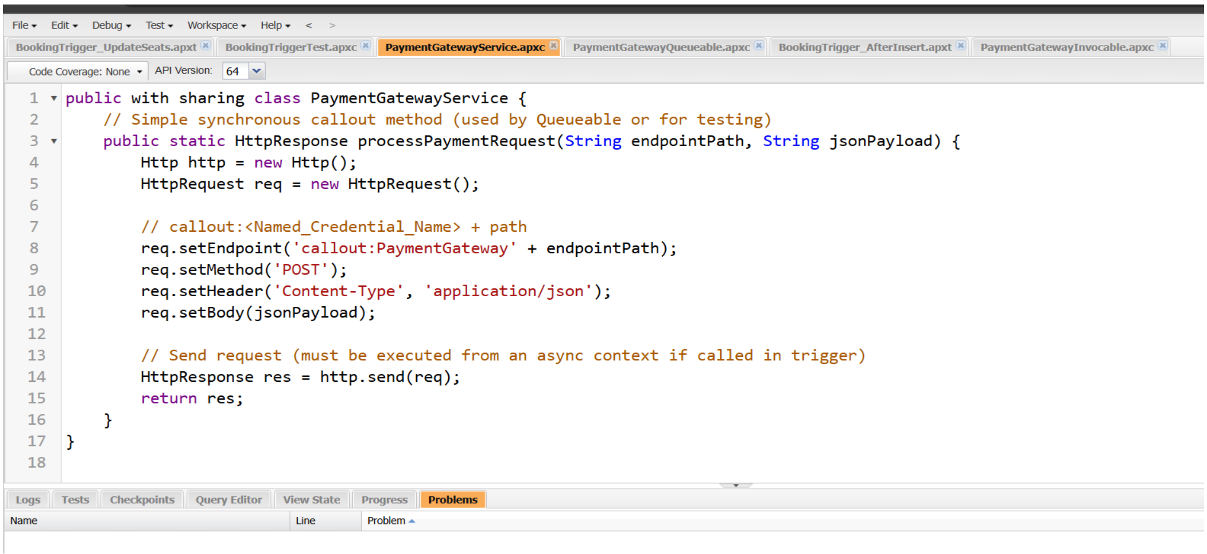
❖ Callouts:

- Implemented HTTP callouts in PaymentGatewayService Apex class to send booking details (Booking Id and total payment) to the payment gateway.
- Received and logged API responses.



```
public with sharing class PaymentGatewayService {
    // Simple synchronous callout method (used by Queueable or for testing)
    public static HttpResponse processPaymentRequest(String endpointPath, String jsonPayload) {
        Http http = new Http();
        HttpRequest req = new HttpRequest();

        // callout:<Named_Credential_Name> + path
        req.setEndpoint('callout:PaymentGateway' + endpointPath);
        req.setMethod('POST');
        req.setHeader('Content-Type', 'application/json');
        req.setBody(jsonPayload);

        // Send request (must be executed from an async context if called in trigger)
        HttpResponse res = http.send(req);
        return res;
    }
}

```

❖ Payment Gateway Service API


File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >
BookingTrigger_UpdateSeats.apxt
BookingTriggerTest.apxc
PaymentGatewayService.apxc
PaymentGatewayQueueable.apxc
BookingTrigger_AfterInsert.apxt
PaymentGatewayInvocable.apxc
Logs
Tests
Checkpoints
Query Editor
View State
Progress
Problems
Name
Line
Problem

Code Coverage: None ▾ API Version: 64

```
public with sharing class PaymentGatewayService {
    // Simple synchronous callout method (used by Queueable or for testing)
    public static HttpResponse processPaymentRequest(String endpointPath, String jsonPayload) {
        Http http = new Http();
        HttpRequest req = new HttpRequest();

        // callout:<Named_Credential_Name> + path
        req.setEndpoint('callout:PaymentGateway' + endpointPath);
        req.setMethod('POST');
        req.setHeader('Content-Type', 'application/json');
        req.setBody(jsonPayload);

        // Send request (must be executed from an async context if called in trigger)
        HttpResponse res = http.send(req);
        return res;
    }
}
```

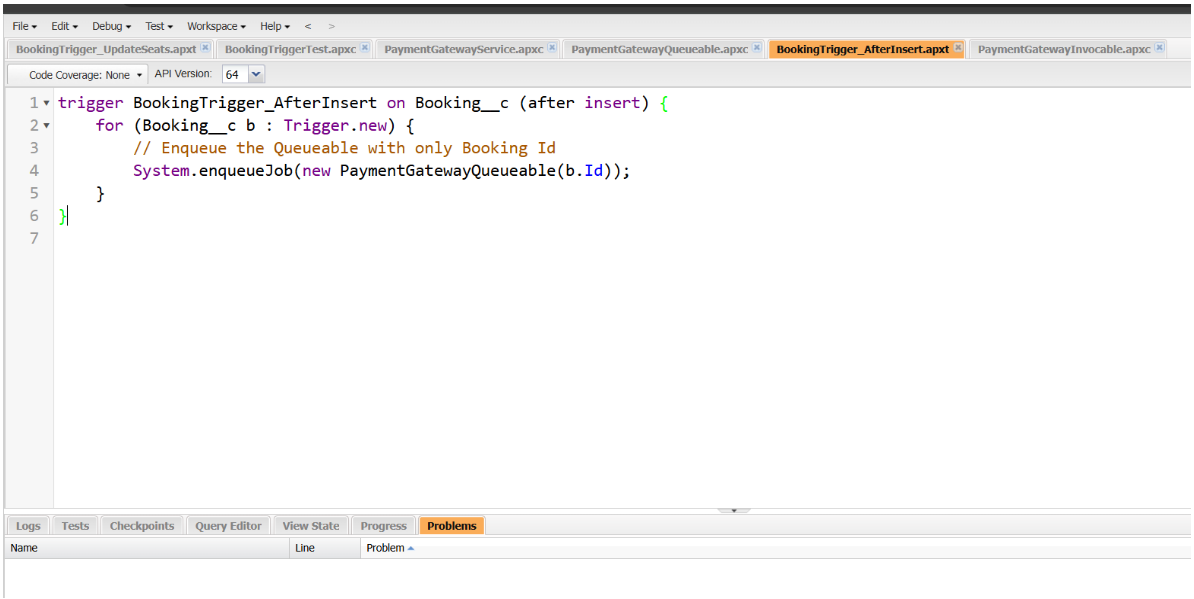
❖ Queueable Apex:

- Created PaymentGatewayQueueable class to handle API callouts asynchronously, keeping Salesforce responsive.
- Queries the Booking record to get the total payment amount before sending it to the API.

```
public class PaymentGatewayQueueable implements System.Queueable, Database.AllowsCallouts {
    private String bookingId;


    // Constructor - just pass Booking Id
    public PaymentGatewayQueueable(Id bookingId) {
        this.bookingId = bookingId;
    }


    public void execute(System.QueueableContext ctx) {
        try {
            // Query Booking to get Total Payment
            Booking__c b = [SELECT Id, Total_Payment__c FROM Booking__c WHERE Id = :bookingId LIMIT 1];


            // Prepare JSON payload
            Map<String, Object> payloadMap = new Map<String, Object>{
                'bookingId' => b.Id,
                'amount'    => b.Total_Payment__c
            };
            String jsonPayload = JSON.serialize(payloadMap);


            // Call the payment API
            HttpResponse res =
PaymentGatewayService.processPaymentRequest('/processPayment',
jsonPayload);
            System.debug('Payment API response: ' + res.getBody());


        } catch (Exception e) {
            System.debug('Payment callout failed: ' + e.getMessage());
        }
    }
}
```

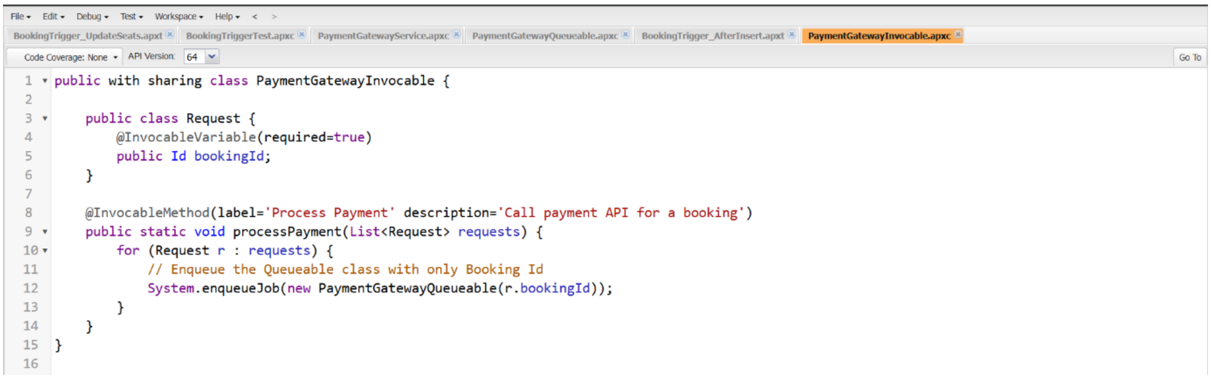
❖ **Trigger on Booking__c:**

- Created an after-insert trigger on the Booking object.
- Automatically enqueues the Queueable class whenever a new booking is created, automating payment processing.



```
trigger BookingTrigger_AfterInsert on Booking__c (after insert) {
    for (Booking__c b : Trigger.new) {
        // Enqueue the Queueable with only Booking Id
        System.enqueueJob(new PaymentGatewayQueueable(b.Id));
    }
}
```

❖ PaymentGatewayInvocable


File ▾
Edit ▾
Debug ▾
Test ▾
Workspace ▾
Help ▾
<
>
BookingTrigger_UpdateSeats.apxt
BookingTriggerTest.apxc
PaymentGatewayService.apxc
PaymentGatewayQueueable.apxc
BookingTrigger_AfterInsert.apxt
PaymentGatewayInvocable.apxc
Go To

Code Coverage: None ▾ API Version: 64

```
public with sharing class PaymentGatewayInvocable {

    public class Request {
        @InvocableVariable(required=true)
        public Id bookingId;
    }

    @InvocableMethod(label='Process Payment' description='Call payment API for a booking')
    public static void processPayment(List<Request> requests) {
        for (Request r : requests) {
            // Enqueue the Queueable class with only Booking Id
            System.enqueueJob(new PaymentGatewayQueueable(r.bookingId));
        }
    }
}

```