

LearnFrenzy

Salesforce Integration Interview Scenarios

2026

SCENARIO 1

Trigger-Based Callout API

CHALLENGE

"When an Order record is inserted, send its data to an external ERP via REST API."

PROPOSED SOLUTION

I created a trigger on Order calling a helper class using Queueable Apex with callout=true.

```
|  
public class OrderCallout implements Queueable, Database.AllowsCallouts {  
    Order__c ord;  
    public OrderCallout(Order__c o) { this.ord = o; }  
  
    public void execute(QueueableContext context) {  
        HttpRequest req = new HttpRequest();  
        req.setEndpoint('https://erp.example.com/api/orders');  
        req.setMethod('POST');  
        req.setHeader('Content-Type', 'application/json');  
        req.setBody(JSON.serialize(ord));  
        Http http = new Http();  
        HttpResponse res = http.send(req);  
    }  
}
```

APEX

BEST PRACTICES

- Enqueue callout outside the trigger loop.
- Use Custom Metadata for API endpoints.

Expose REST API

CHALLENGE

"How would you let external systems create Contact records safely?"

PROPOSED SOLUTION

I exposed an Apex class using `@RestResource` and `@HttpPost`.

```
|  
@RestResource(urlMapping='/createContactAPI/*')  
global with sharing class ContactAPI {  
    @HttpPost  
    global static String createContact() {  
        RestRequest req = RestContext.request;  
        String body = req.requestBody.toString();  
        Contact c = (Contact) JSON.deserialize(body, Contact.class);  
        insert c;  
        return 'Contact Created: ' + c.Id;  
    }  
}
```

APEX

BEST PRACTICES

- Secure with Auth Provider/Named Credentials.
- Use try-catch for error handling.

SCENARIO 3

Flow External Service

CHALLENGE

"Can you invoke an API from Flow without Apex?"

PROPOSED SOLUTION

Yes, using External Services with OpenAPI Schema via Flow Action.

```
|
paths:
  /convertcurrency:
    post:
      summary: Convert Currency
      requestBody:
        content:
          application/json:
            schema:
              type: object
              properties:
                amount: { type: number }
                fromCurrency: { type: string }
      responses:
        '200': { description: Conversion Result }
```

YAML

BEST PRACTICES

- No-code solution for admins.
- Register schema via Named Credentials.

SCENARIO 4

Consume SOAP Service

CHALLENGE

"How do you consume a legacy SOAP Service?"

PROPOSED SOLUTION

I used WSDL2Apex to generate classes and invoked standard SOAP methods.

```
LegacyServicePort svc = new LegacyServicePort();
svc.inputHttpHeaders_x = new Map<String, String>();
svc.inputHttpHeaders_x.put('SOAPAction', 'createCustomer');

String res = svc.createCustomer('John', 'Doe', 'j@d.com');
System.debug('Response: ' + res);
```

APEX

BEST PRACTICES

- Used in legacy/banking systems.
- Always mock WSDL in test classes.

Retry Logic

CHALLENGE

"How do you retry failed API requests?"

PROPOSED SOLUTION

I logged failed requests in a custom object and used a Scheduled Job to retry.

```
global class RetryFailedCallouts implements Schedulable {
    global void execute(SchedulableContext sc) {
        List<Failed_Integration__c> fList = [SELECT Id, Endpoint__c, Payload__c
            FROM Failed_Integration__c WHERE Status__c = 'Failed'];

        for(Failed_Integration__c f : fList){
            HttpRequest req = new HttpRequest();
            req.setEndpoint(f.Endpoint__c);
            req.setMethod('POST');
            req.setBody(f.Payload__c);
            try {
                Http http = new Http();
                HttpResponse res = http.send(req);
                if(res.getStatusCode() == 200) f.Status__c = 'Success';
            } catch(Exception e){ f.Retry_Count__c += 1; }
        }
        update fList;
    }
}
```

APEX

BEST PRACTICES

- Use exponential backoff logic.
- Track retry count and error logs.