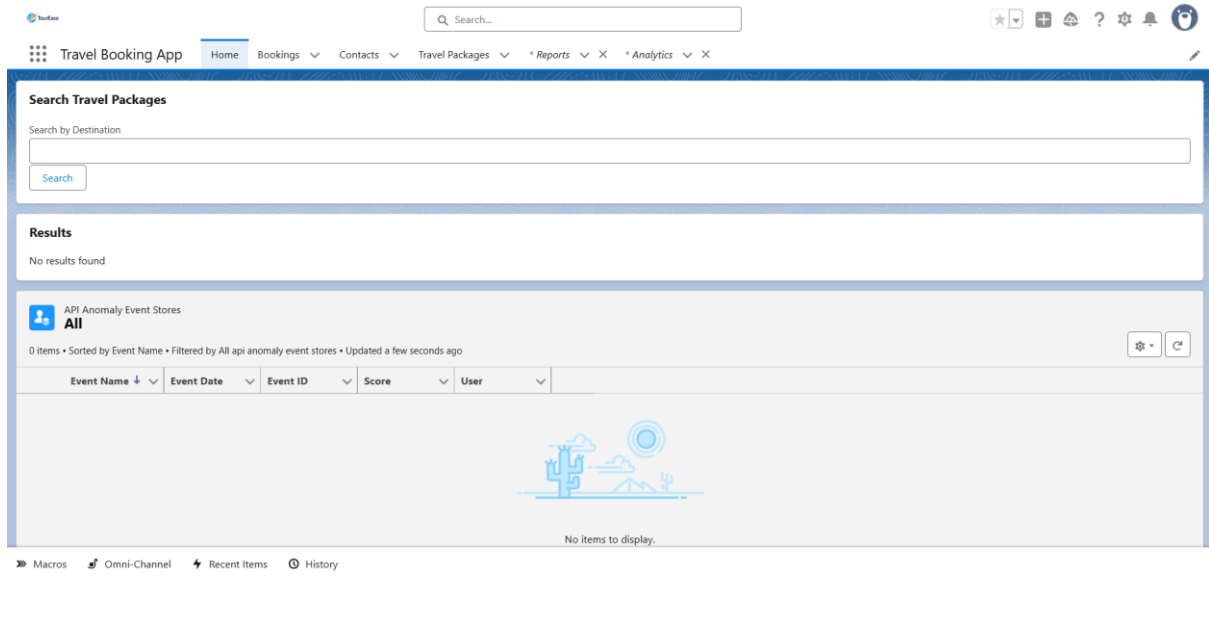


## Phase 7: Integration & External Access

### Objective

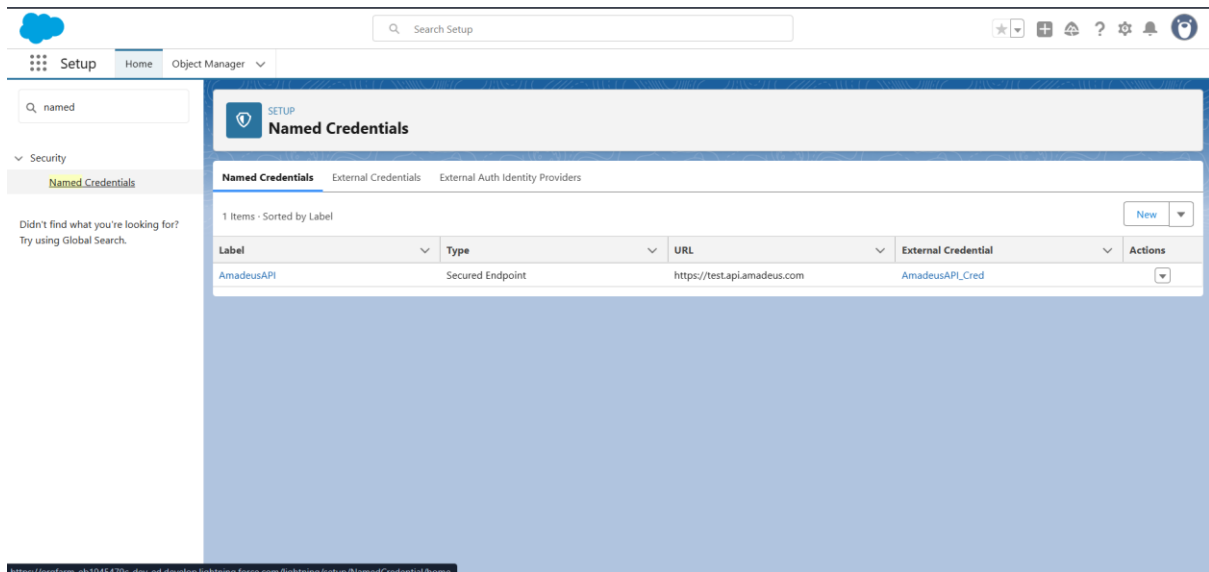
The goal of Phase 7 is to integrate Salesforce with external Travel APIs and Booking Systems to achieve seamless synchronization of travel packages, bookings, and customer records. This phase also ensures secure, real-time communication between systems through APIs, callouts, and platform events.



### 1. Named Credential

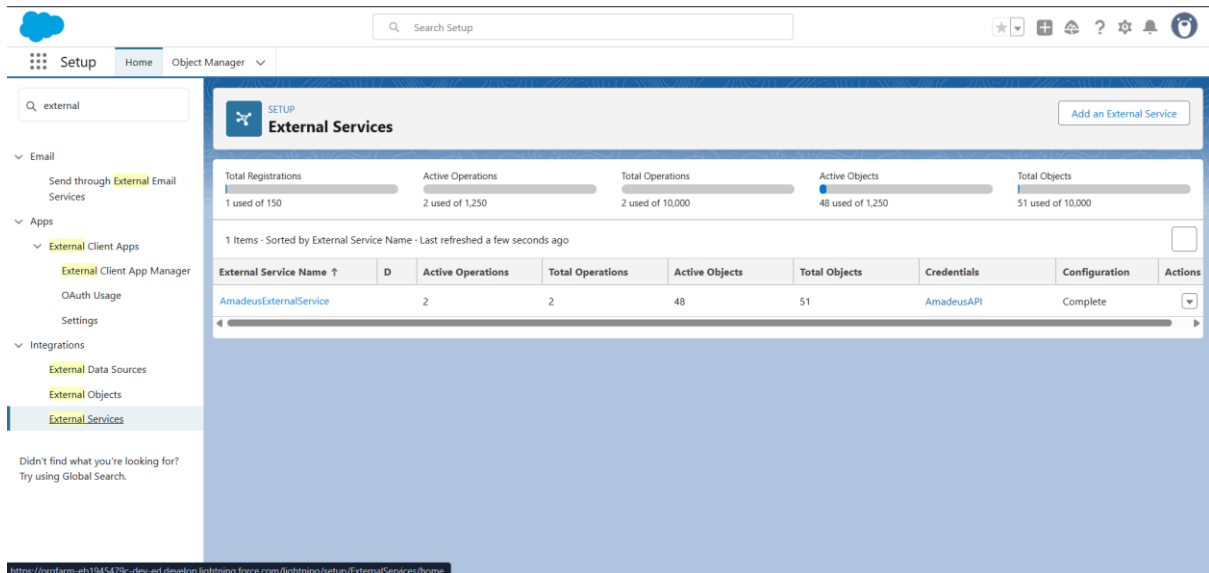
Setup → Named Credentials → New

- **Label:** AmadeusAPI
- **URL:** <https://test.api.amadeus.com/>
- **Identity Type:** Secured Endpoint



## 2. External Services

- **Purpose:** Connect Salesforce to external REST APIs declaratively using OpenAPI specifications.
- **Setup:** Setup → External Services → New External Service
  - Register your Travel API Swagger/OpenAPI definition
  - Automatically generates Apex actions for Flow and Process Builder



## 3. Apex Class: Outbound Callout

Apex Queueable class pushes new bookings to the external Travel API.

public with sharing class TravelBookingPushJob implements Queueable,  
Database.AllowsCallouts {

private Id bookingId;

```

public TravelBookingPushJob(Id bookingId){ this.bookingId = bookingId; }

public void execute(QueueableContext ctx) {
    Booking__c b = [
        SELECT Id, Name, Travel_Package__r.Package_Name__c,
            Customer__r.Email, Booking_Date__c
        FROM Booking__c WHERE Id = :bookingId LIMIT 1
    ];

    Map<String,Object> payload = new Map<String,Object>{
        'customerEmail' => b.Customer__r.Email,
        'packageName'  => b.Travel_Package__r.Package_Name__c,
        'bookingDate'  => String.valueOf(b.Booking_Date__c),
        'bookingId'    => b.Id
    };

    HttpRequest req = new HttpRequest();
    req.setEndpoint('callout:TravelAPI_NC/api/bookings');
    req.setMethod('POST');
    req.setHeader('Content-Type','application/json');
    req.setBody(JSON.serialize(payload));

    Http http = new Http();
    HttpResponse res = http.send(req);
    System.debug('Response: ' + res.getBody());
}
}

```

---

## 4.Callouts

- **Purpose:** Make HTTP requests to external APIs from Apex.
- **Setup:** Use @future(callout=true) or Queueable Apex for asynchronous callouts.
- **Example:** TravelBookingPushJob sends new bookings to external Travel API.

- Code:

```
public with sharing class TravelBookingPushJob implements Queueable,
Database.AllowsCallouts {
```

```
    private Id bookingId;
```

```
    // Constructor to pass Booking Id
```

```
    public TravelBookingPushJob(Id bookingId){
        this.bookingId = bookingId;
    }
```

```
    // Main execution method
```

```
    public void execute(QueueableContext ctx) {
```

```
        // Query Booking record with related Customer and Travel Package
```

```
        Booking__c booking = [
            SELECT Id, Name, Booking_Date__c,
                Travel_Package__r.Package_Name__c,
                Customer__r.Email
            FROM Booking__c
            WHERE Id = :bookingId
            LIMIT 1
        ];
```

```
        // Prepare payload to send to external API
```

```
        Map<String, Object> payload = new Map<String, Object>{
            'bookingId' => booking.Id,
            'customerEmail' => booking.Customer__r.Email,
            'packageName' => booking.Travel_Package__r.Package_Name__c,
            'bookingDate' => String.valueOf(booking.Booking_Date__c)
        };
```

```
        // Create HTTP request
```

```

HttpRequest req = new HttpRequest();
req.setEndpoint('callout:TravelAPI_NC/api/bookings'); // Named Credential endpoint
req.setMethod('POST');
req.setHeader('Content-Type', 'application/json');
req.setBody(JSON.serialize(payload));

// Send HTTP request
Http http = new Http();
try {
    HttpResponse res = http.send(req);
    System.debug('Travel API Response: ' + res.getBody());
} catch (Exception e) {
    System.debug('Error sending booking to Travel API: ' + e.getMessage());
}
}
}

```

---

## 5. Apex REST Service: Inbound Updates

Handles updates from external Travel API about booking confirmations or cancellations.

```

@RestResource(urlMapping='/travel/bookings/*')
global with sharing class TravelBookingWebhook {
    @HttpPost
    global static void receiveUpdate() {
        RestRequest req = RestContext.request;
        Map<String,Object> payload =
            (Map<String,Object>) JSON.deserializeUntyped(req.requestBody.toString());

        String bookingId = (String) payload.get('bookingId');
        String status = (String) payload.get('status');

        Booking__c b = [SELECT Id FROM Booking__c WHERE Id = :bookingId LIMIT 1];
        b.Status__c = status;
        update b;
    }
}

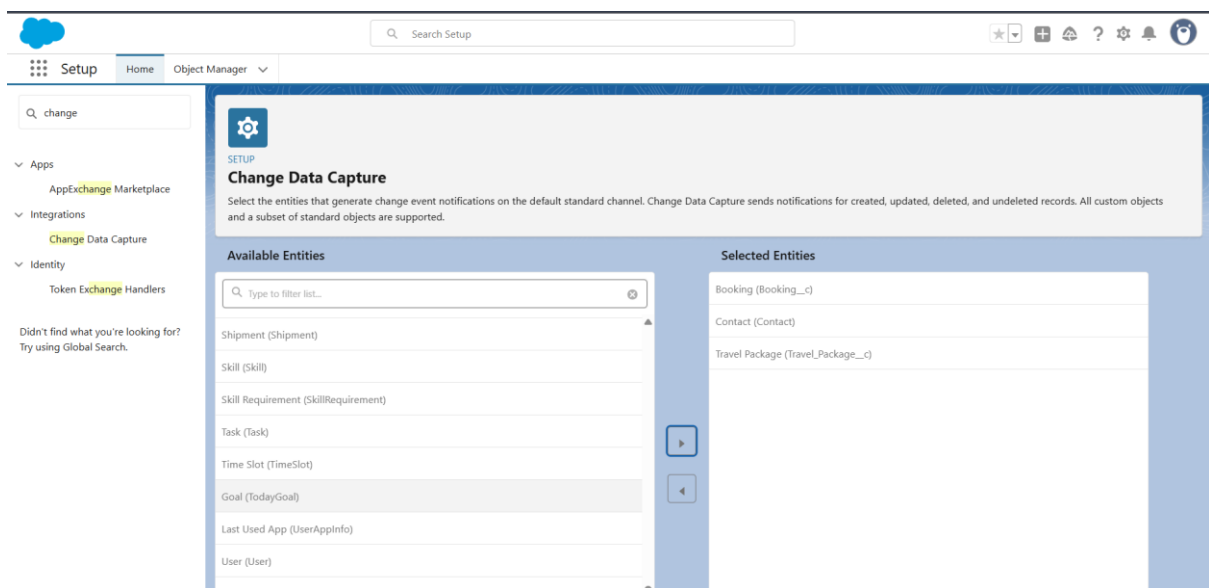
```

```
    RestContext.response.statusCode = 200;
  }
}
```

---

## 6. Change Data Capture (CDC)

- **Purpose:** Automatically capture changes on objects and notify external systems.
- **Setup:** Setup → Change Data Capture → Enable for Booking\_\_c, Travel\_Package\_\_c.
- Works with CometD / Streaming API for real-time integrations.

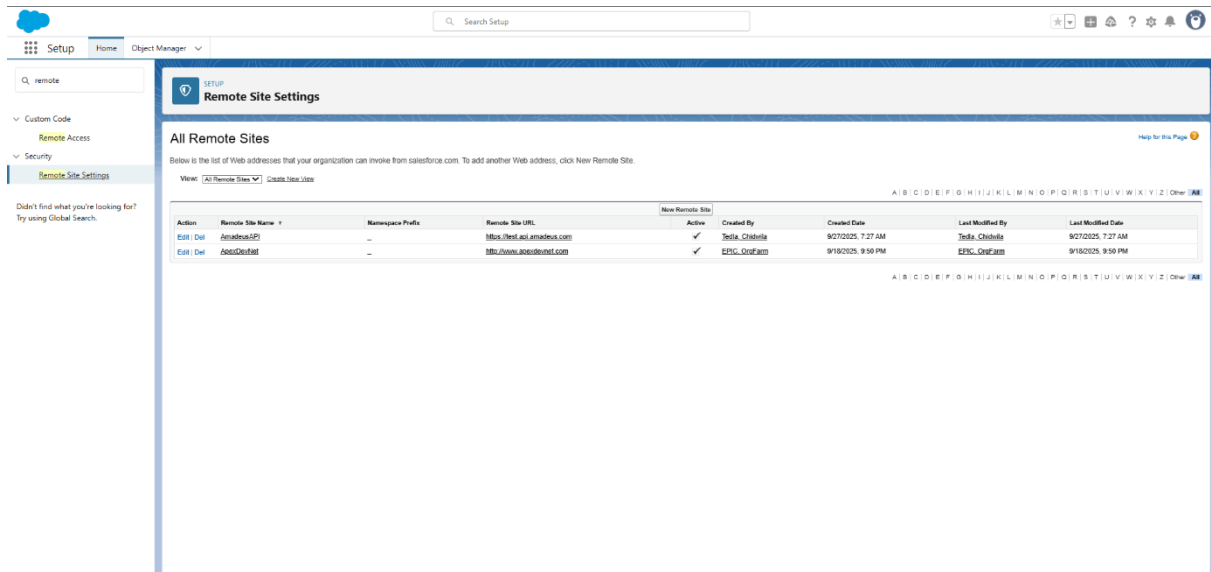


## 7. Salesforce Connect

- **Purpose:** Access external data in real-time without storing it in Salesforce.
- **Setup:** Setup → Salesforce Connect → External Data Source → New.



- Add `https://api.travelexample.com` to allow callouts.



The screenshot shows the Salesforce Setup interface. The left sidebar contains a search bar with 'remote' entered, and a list of setup categories: Custom Code, Remote Access, Security, and Remote Site Settings (highlighted). The main content area is titled 'Remote Site Settings' and includes a sub-header 'All Remote Sites'. Below this, a message states: 'Below is the list of Web addresses that your organization can invoke from salesforce.com. To add another Web address, click New Remote Site.' There are two tabs: 'All Remote Sites' (selected) and 'Create New Site'. A table lists the remote sites with columns for Action, Remote Site Name, Namespace Prefix, Remote Site URL, Active status, Created By, Created Date, Last Modified By, and Last Modified Date. Two sites are listed: 'AmadeusSP' and 'AmadeusNet'. Both are active and created by 'Teda Chabela' on 9/27/2025 at 7:27 AM. The 'AmadeusNet' site has a last modified date of 9/18/2025 at 9:50 PM by 'EPNC\_OrgTeam'. Alphabetical filters are visible at the top and bottom of the table.

Action	Remote Site Name	NameSpace Prefix	Remote Site URL	Active	Created By	Created Date	Last Modified By	Last Modified Date
<a href="#">Edit</a> <a href="#">Del</a>	AmadeusSP	--	https://test.api.amadeus.com	✓	Teda Chabela	9/27/2025, 7:27 AM	Teda Chabela	9/27/2025, 7:27 AM
<a href="#">Edit</a> <a href="#">Del</a>	AmadeusNet	--	https://www.amadeusnet.com	✓	EPNC_OrgTeam	9/18/2025, 9:50 PM	EPNC_OrgTeam	9/18/2025, 9:50 PM

## Conclusion

Phase 7 ensures secure, real-time integration between Salesforce and external travel systems. Named Credentials, External Services, REST/SOAP callouts, Platform Events, CDC, and API management provide a robust enterprise-grade integration layer for TourEase.