Phase 7: Integration & External Access

Travel and Tourism Management System (TTMS)

1. Named Credential Configuration

Purpose:

Securely manage API credentials and endpoint URLs, avoiding hardcoding sensitive information in code.

Configuration:

Label: SmartTripAPI

• URL: https://api.smarttrip.com/v1

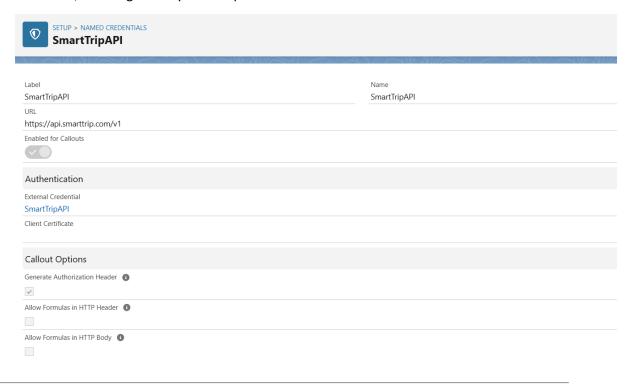
• Authentication: Linked to External Credential or relevant auth mechanism

Options:

- Enabled for Callouts
- Generate Authorization Header (checked)

• Sample Salesforce View:

The Named Credential SmartTripAPI is used as a reference for all future API callouts in Apex and Flows, ensuring security and simplified maintenance.



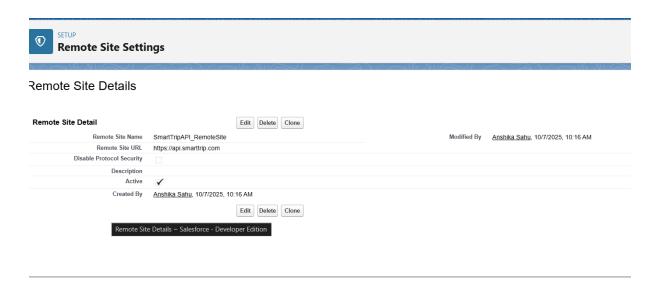
2. Remote Site Settings

Purpose:

Authorize Salesforce org to make HTTP callouts to external domains.

Configuration:

- Name: SmartTripAPI_RemoteSite
- URL: https://api.smarttrip.com/v1
- Step:
 - Setup → Remote Site Settings → New → Fill in Name & URL → Save



3. REST Callouts (Apex Example)

Purpose:

Programmatically fetch or send data to the SmartTrip external service.

Apex Example:

```
text
public with sharing class SmartTripBookingService {
   public static String getBookings() {
      HttpRequest req = new HttpRequest();
      req.setEndpoint('callout:SmartTripAPI/bookings');
      req.setMethod('GET');
      Http http = new Http();
      HttpResponse res = http.send(req);
      System.debug(res.getBody());
      return res.getBody();
   }
}
```

• Best Practice:

- Always use the callout:SmartTripAPI/... format in endpoints.
- Handle errors and parse responses in production code.

4. OAuth & Authentication

Purpose:

Establish secure and delegated access to third-party systems using standard OAuth 2.0 protocol.

Configuration:

- Register an OAuth Provider in Salesforce Setup.
- Input client credentials from the vendor.
- Link this Auth Provider to the SmartTripAPI Named Credential for token management.
- Test authentication/authorization handshake before production usage.

5. External Services (Optional: For Declarative Integration)

Purpose:

Expose external APIs as usable Flow actions via OpenAPI schema registration.

Step:

- Obtain OpenAPI/Swagger specification from the external service.
- Setup → External Services → New
- Choose SmartTripAPI as Named Credential, enter schema URL.
- Actions from the schema become available as Flow elements.

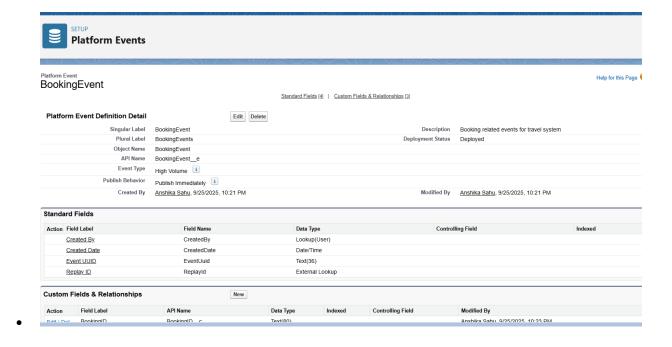
6. Platform Events

Purpose:

Send real-time notifications of important changes, such as booking confirmations/cancellations, to interested Salesforce apps or external listeners.

Configuration:

- Setup → Platform Events → New
 - Example Event: BookingEvent__e
 - Fields: BookingID, CustomerID, Status, TimeStamp
- Publish Platform Events via Apex trigger or Flow when a booking lifecycle changes.



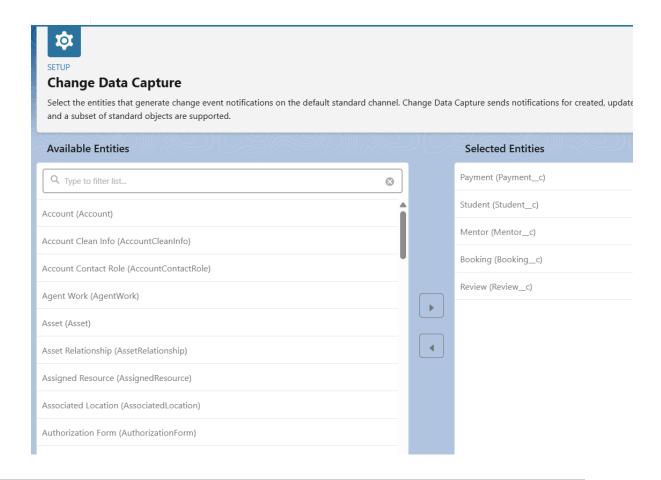
7. Change Data Capture (CDC)

Purpose:

Enable real-time capture of changes (create/update/delete) on Booking, Customer, or Payment objects for sync with external systems.

Configuration:

- Setup → Change Data Capture
- Enable CDC for relevant objects
- Use Streaming API for external systems to subscribe to change events.



8. Salesforce Connect

Purpose:

Access external travel data in Salesforce in real-time without duplication.

Configuration:

- Setup → External Data Sources → New
- Type: OData 2.0/3.0
- URL: Provided by travel partner/service
- Authenticate via Named Credential
- Sync objects, use directly in reports and relationships.

9. API Usage Limits

Purpose:

Respect Salesforce org's API daily/concurrent usage limitations.

Practice:

- Monitor via Setup → System Overview
- Code for graceful error handling and comply with limits.

• Use bulk or asynchronous processing for heavy operations.

Summary

By following these steps and using Salesforce's secure integration patterns, the Travel and Tourism Management System will achieve robust and scalable connectivity to SmartTripAPI and other partner APIs:

- Sensitive data is protected.
- Callouts and data sync are managed declaratively and programmatically.
- Real-time events and data changes are communicated to internal/external consumers.
- Future service integrations are simplified with maintained, modular architecture.