What is soap?

1.Simple Object Access protocols Relies on XML and schema

2.Strongly types messaging framework

3.Operations and its XML structure of request and response defined

4.Communicated through WSDL(Contract between both)

What Steps to follow for SOAP Integration?

1.Create Webservice Method

2.Generate and share wsdl with client

3.Client generate apex class from wsdl

4.Invoke soap call to server.

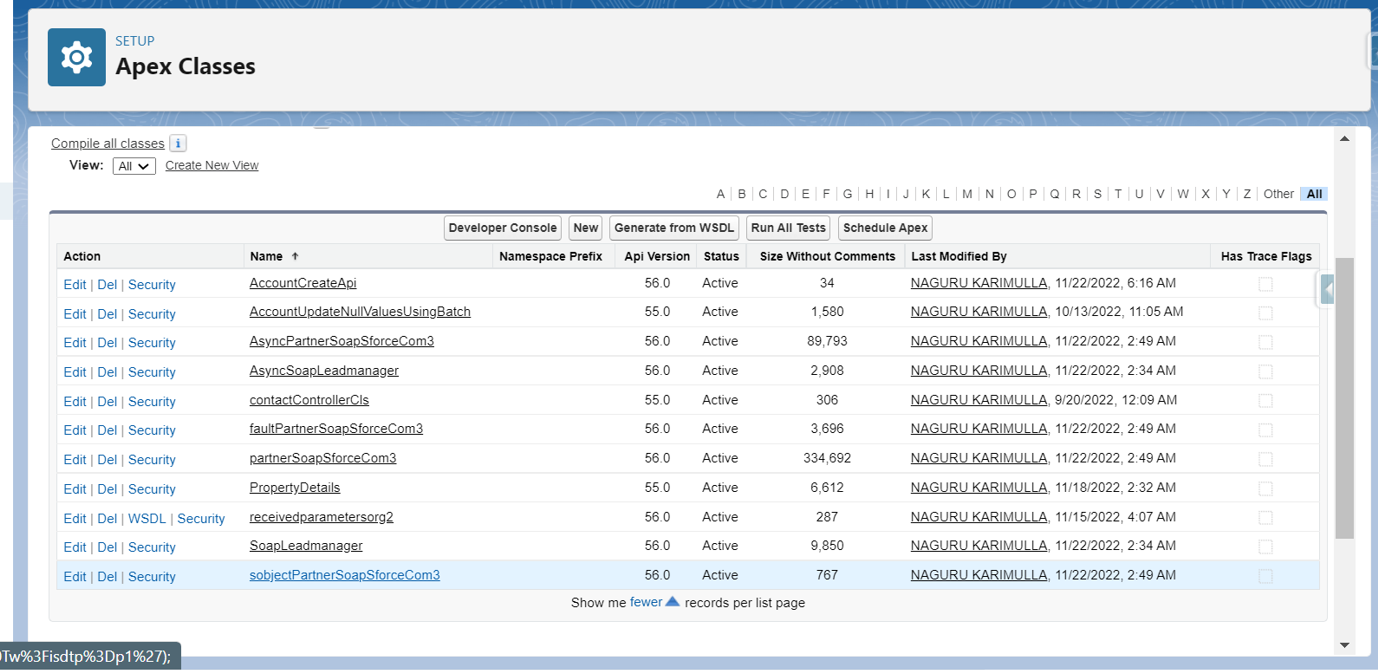
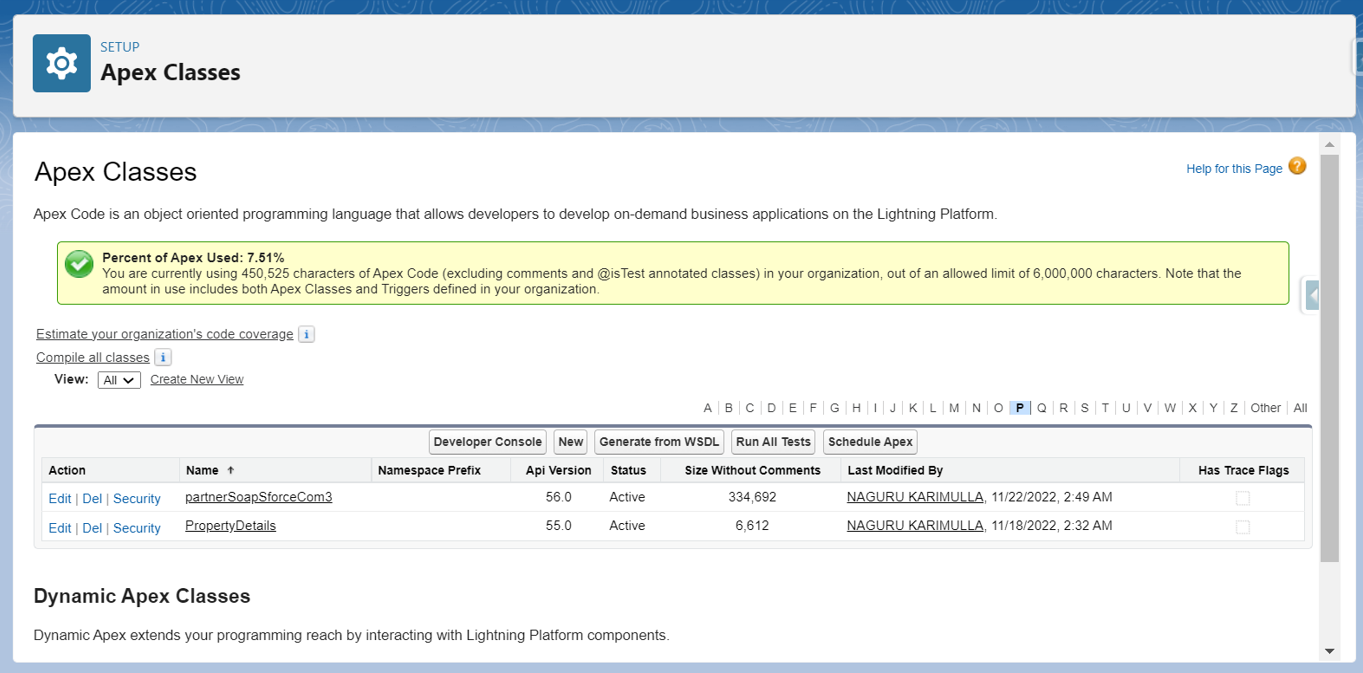
Steps To Expose SOAP Services

1. Create a Apex class
2. Add Webservice method
3. Complete your server logic
4. Expose as WSDL

Soap Integration

Apex class method exposed as custom SOAP web service calls

1. Use webservice keyword
2. Web service method always use system context
3. Class contain webservice method declare as global
4. Define method as static

Rest vs Soap

**REST allows a greater variety of data formats, whereas SOAP only allows XML**. Coupled with JSON (which typically works better with data and offers faster parsing), REST is generally considered easier to work with. Thanks to JSON, REST offers better support for browser clients.

Why is SOAP More Secure? Although SOAP and REST both support SSL (Secure Socket Layer) for data protection, while making the request, **SOAP supports Web Services Security (also known as WS- Security or WSS) for enterprise-level protection which is absent in REST Services**.

**REST is a better choice for simple, CRUD-oriented services**, because of the way REST repurposes HTTP methods (GET, POST, PUT, and DELETE). It is also popular because it's lightweight and has a smaller learning curve. SOAP, on the other hand, has standards for security, addressing, etc.

API described in Salesforce documentation.  
**SOAP API**  
Supports data in the form of XML only  
Requires WSDL for the integration  
Use SOAP API in any language that supports Web services.  
It is a standard protocol and needs more bandwidth and resources.  
  
**REST API**  
Supports both XML and JSON format  
Preferred for mobile and web apps since JSON being Lighter the app runs smoother and faster  
It is more of an architectural system and requires less bandwidth and resources.

Connected App

1. is a framework that enables an external application to integrate with Salesforce using APIs and standard protocols, such as SAML, OAuth, and OpenID Connect to authenticate, authorize, and provide single sign-on (SSO) for external apps.
2. When to use connected app
3. Access data with API Integration
4. Integrate service provider with salesforce org
5. Manage access to third party apps
6. Provide authorization for external api

Create a connected app

Define basic like app name, logo, contact information

-Setup-App Manager –Connected App

use a connected app to request access to Salesforce data on the behalf of an external application

-must be integrated with the Salesforce API using the OAuth 2.0 protocol

-select**Enable OAuth Settings**

-Enter the callback URL

-Select OAuth scopes

Rest

API: set of rules that allow programs to talk each other

-Developer created API on server and allows client to call it

-REST-Representational State Transfer-is a set of rules that developer follow when they create API

Rest request in salesforce

Access data using RESTFull pattern

XML or JSON

Each resource in REST API is identified by a named Uniform Resource Identifier(URI)

accessed using standard HTTP methods

Anatomy of rest request:

Endpoint

-Rest Methods

-Rest Headers

-Rest Body

Rest methods

GET–Retrieve data

POST -Create a resource or post data to the server

PUT -Create or replace the resource sent in the request body

DELETE-Delete a resource identified by a URL

PATCH-Update all the representations of the member resource

Rest in apex

* 1. Use @RestResource annotation in classDefine URL mapping on class level
  2. Wild charcter \* can use
  3. URL Mapping case sensitive
  4. Apex class defined as global
  5. Cannot have multiple method with same annotations
  6. -Use below annotation in methods @httpget ---No parameter
  7. @httpPost
  8. @httpPut
  9. @httpPatch
  10. @httpDelete –No parameter