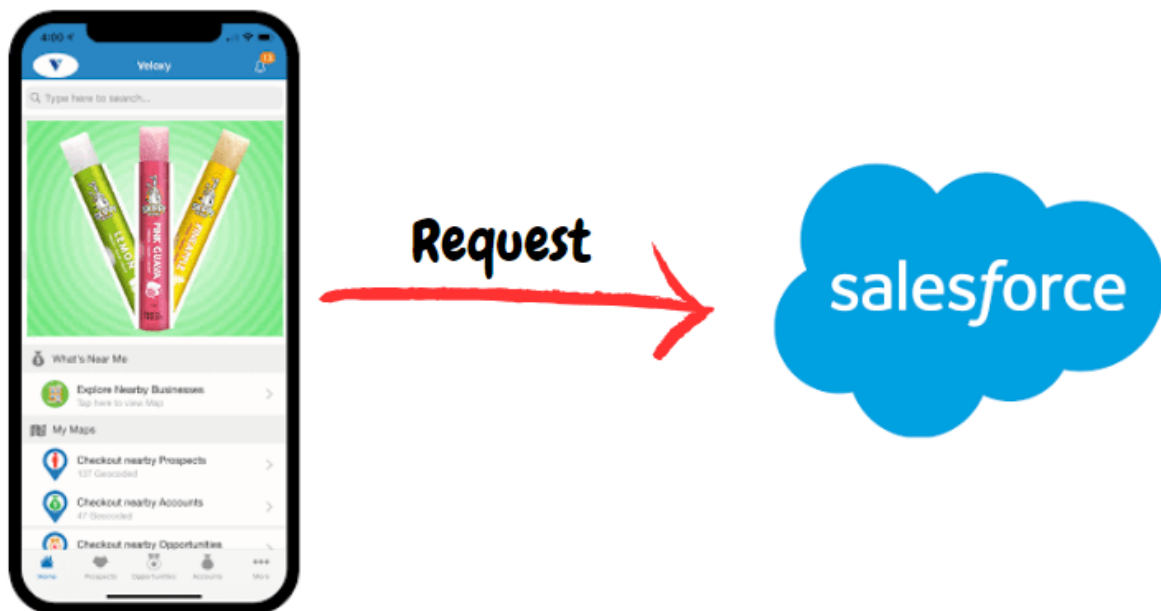**By Smriti Sharan (sfdcamplified)**

Complete Guide with 80+ Salesforce Integration Interview Questions Covered

Hello, my Name is Smriti Sharan. I am avid blogger and youtuber. **Follow my Blog and youtube to learn various aspect of Salesforce.**

https://t.me/sfdcamplified feel free to join telegram group.

1.Explain the difference between inbound and outbound integration in Salesforce (very important)

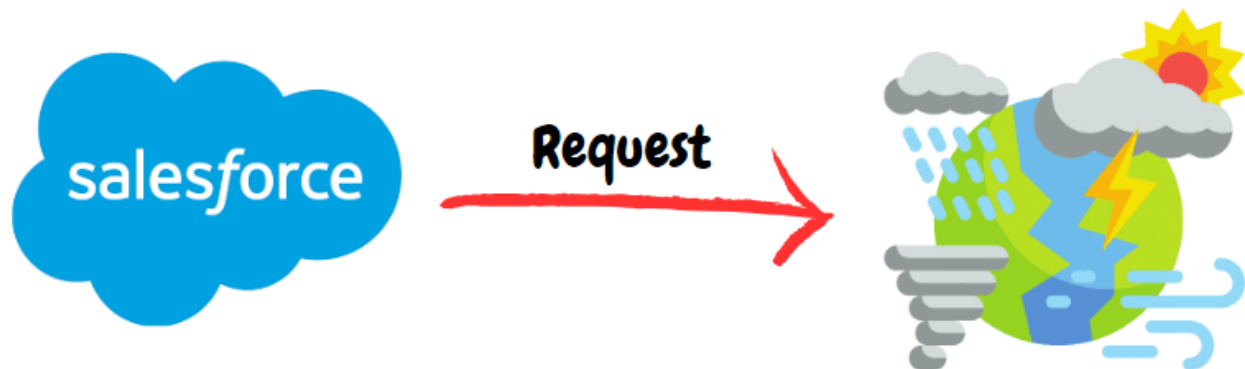Inbound Integration External systems send data to Salesforce. Example An ERP system sends order data to Salesforce.

Outbound Integration Salesforce sends data to external systems.
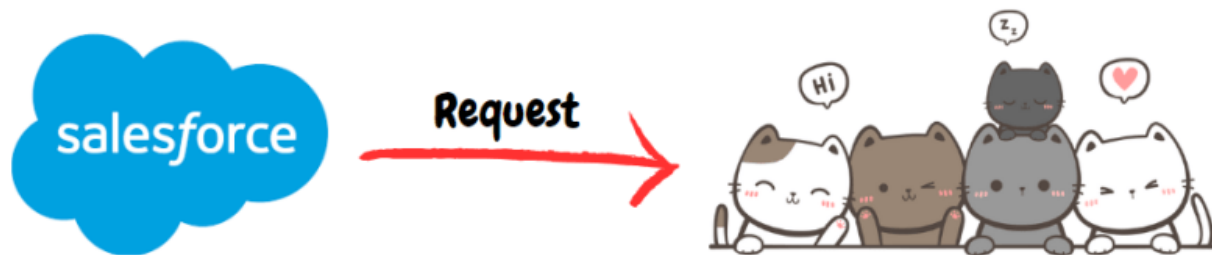
Example:

By Smriti Sharan (sfdcamplified)

When querying a weather API for the latest forecasts

Fetching cute cat pictures from a cat API

2.Difference between authentication and authorization?

Authentication verifies who you are.

By Smriti Sharan ([sfdcamplified](#))

Authorization verifies what you are authorized to do.

What are types of authentication?
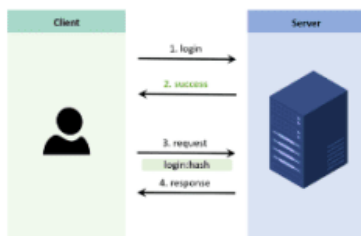
By Smriti Sharan ([sfdcamplified](#))
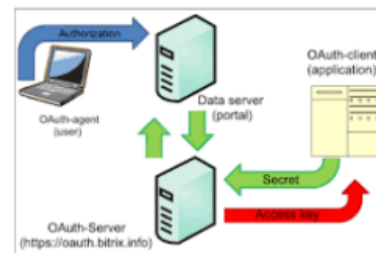
Basic Authentication

API Key Authentication

Digest Authentication

Oauth 2.0

Basic authentication

Basic Authentication is parallel to using a traditional key to unlock a door. It involves sending a username and password with every HTTP request.

API Key

API Key Authentication, in comparison, is like having a personalized access card. It involves a unique key that the server uses to verify the authenticity of a request.

OAuth

OAuth is a more complex protocol designed for scenarios where you delegate access to your Salesforce data to third-party applications without exposing your credentials. It involves a multi-step process like obtaining a request token, getting user authorization, and then exchanging the token for an access token.

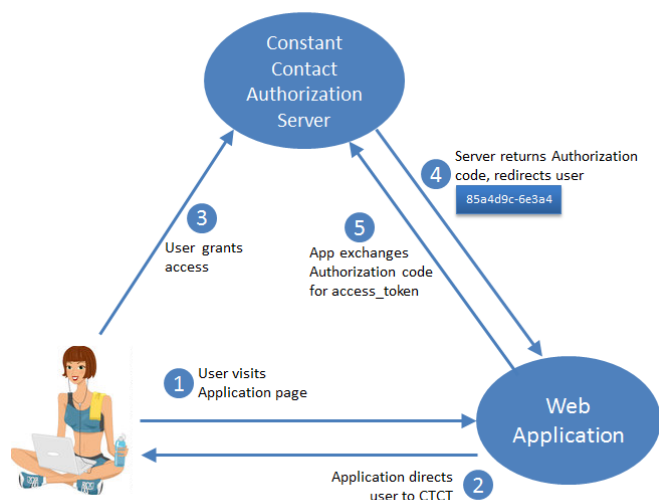**3.What are type of flows supported by Salesforce? (very important question in interview)**

1. Web Server Authentication

2. User-Agent

3. JWT Bearer Token

4. SAML Bearer Assertion

5. SAML Assertion

6. Username-Password

7. Device Authentication

8. Asset Token

9. Refresh Token

4.Explain OAuth 2.0 web server flow ? (very important)

By Smriti Sharan (sfdcamplified)

OAuth 2.0 Web Server Flow also known as the Authorization Code Grant Flow.

1. Client Application Logs in to Salesforce. The client application directs the user to Salesforce's authorization endpoint using the API user credentials.

2. Salesforce Authorization Server Verifies the Request. Salesforce's authorization server verifies the request parameters and prompts the user to log in if not already authenticated.

3. Authorization Server Sends Authorization Code. After the user authorizes the application, Salesforce's authorization server sends an authorization code to the client application's redirect URI.

4. Client Application Requests Access Token. The client application sends a POST request to Salesforce's token endpoint (/services/oauth2/token) to exchange the authorization code for an access token.

The request includes parameters like grant_type=authorization_code, client_id, client_secret, redirect_uri, and the code received in the previous step.

5. Authorization Server Verifies Client ID and Secret. Salesforce's authorization server verifies the provided client_id, client_secret, redirect_uri, and the authorization code.

6. Authorization Server Sends Access Token. Upon successful verification, Salesforce's <mark>authorization server responds with an access token</mark>.

7.The client application can now use the access token to make authenticated requests to Salesforce APIs.

8.Salesforce's API receives the request, validates the access token, and processes the request according to the specified endpoint and HTTP method (GET, POST, PATCH, DELETE, etc.).

5.Explain Steps of OAuth 2.0 User Agent Flow?

OAuth 2.0 User Agent Flow (also known as the <mark>Implicit Grant Flow</mark>) is used for <mark>client-side applications</mark> (e.g.single-page applications) where the client secret cannot be securely stored.

1.The <mark>client application redirects</mark> the user's browser to Salesforce's authorization endpoint.

2.Salesforce's authorization server prompts the user to log in if not already authenticated.

3. After logging in, the user is prompted to authorize the client application to access their Salesforce data.

4. Upon user authorization, Salesforce's authorization server redirects the user's browser to the client application's redirect URI. The access token is included in the fragment of the URL.

5. Client Application Extracts Access Token

6. Client Application Makes Authenticated API Requests.

7. Salesforce's API validates the access token and processes the request.

8. Salesforce responds with the requested data

6.What is composite API?

The Composite API in Salesforce allows you to execute multiple related operations in a single request.

Example: Create a customer, create an order for the customer, add items to the order, and change the order status in single request.

7.What are Common Integration Patterns in Salesforce? (important)

1. Request and Reply

In this pattern, the client application sends a request and waits for a reply before continuing with the next steps. This is a synchronous process where the client expects an immediate response from the server.

Example: Integrating Salesforce with Stripe for payment processing. The Salesforce application waits for a payment response from Stripe before proceeding with further actions. This can be achieved through Apex classes. It does not involve asynchronous methods like batch, future.

2. Fire and Forget

In this pattern, a request is sent ==without waiting for a response==. The process continues immediately after the request is sent.

Example: A record update in Salesforce, triggers the publication of a platform event that AWS or SAP ERP systems can subscribe to.

This is typically achieved using ==Platform Events==.

EventBus.publish(event);

3. Batch Data Synchronization

This pattern is used to ==synchronize a large volume of records== between Salesforce and an external system, typically on a scheduled basis.This is generally achieved using the ==Bulk API.==

4. Remote Call-In

In this pattern, an external system calls into Salesforce to perform operations such as creating, updating, or querying records.

Example: A campaign is created in Sizmek, and the details are sent to Salesforce to create a corresponding campaign record.

This is generally achieved using Salesforce APIs (SOAP/REST).

```
@RestResource(urlMapping='/createCampaign/*')

global with sharing class CampaignService {

    @HttpPost

    }

}
```

5. Data Virtualization

This pattern allows Salesforce to access data stored in external systems ==without physically storing the data within Salesforce.==

Example: Using Salesforce Connect to access account and order data stored in Informatica without replicating the data in Salesforce.

8.How can you achieve realtime integration between Salesforce and an external system for data synchronization?

To achieve realtime integration, we can use Salesforce Change Data Capture (CDC) for capturing changes in Salesforce records and transmitting them to an external system.

9.What is the difference between REST API and SOAP API in Salesforce? (very important for interview)

REST API

- Uses RESTful principles.
- Lightweight and stateless.
- Works with JSON and XML formats.
- Generally used for mobile and web applications.

SOAP API

- Uses SOAP protocol.
- Heavier with more builtin standards for security and transactions.
- Works with XML format.
- Suitable for enterpriselevel integrations requiring robust security.

10.Explain the concept of Salesforce Connect and its use case?

Salesforce Connect allows integration of external data sources with Salesforce, enabling users to view and interact with external data which are not stored in Salesforce.

Implementation Steps

- Go to Setup.
- Enter "External Data Sources" in the Quick Find box.

- Click "New External Data Source" and configure it
- Click "Validate and Sync".
- Select the tables you want to sync.
- The selected tables will be created as external objects.
- Create lookup relationships between Salesforce objects and external objects if necessary.
- Use external objects in Salesforce reports and dashboards as if they were standard objects.

11. What is a Named Credential in Salesforce, and why is it used?

A Named Credential specifies the URL of a callout endpoint and its required authentication parameters in one place.

12. How do you secure API integrations in Salesforce?

- Implement OAuth 2.0 for securing API access.
- Configure Connected Apps for third party systems.
- Ensure all communications use HTTPS to encrypt data in transit.
- Store authentication details in Named Credentials to avoid hardcoding sensitive information.
- Restrict API access to specific IP ranges.
- Create an integration user with minimal permissions required for the integration.
- Monitor API usage and access through Salesforce's audit logs.

13. What is the difference between JSON and XML?

JSON (JavaScript Object Notation)

- Lightweight and easy to read/write.
- Data format Key value pairs.
- Commonly used for REST APIs.
- Better suited for low processing devices.

XML (eXtensible Markup Language)

- More verbose and structured.
- Data format Hierarchical structure with tags.
- Commonly used for SOAP APIs.
- Supports namespaces and schema definitions.

14.What is Streaming API in Salesforce, and when would you use it?

Streaming API is used to ==receive realtime notifications== of changes in Salesforce data. It's ideal for scenarios where you need to keep external systems synchronized with Salesforce changes.

15.How do you implement error handling in Salesforce integration?

Effective error handling in Salesforce integration involves capturing errors, logging them, and implementing retry mechanisms.

1. Apex Callout Error Handling

   Use try catch blocks to handle exceptions during Apex callouts.

```apex
public void makeCallout() {
  try {
      HttpRequest req = new HttpRequest();
      req.setEndpoint('https//api.example.com/data');
      req.setMethod('GET');

      Http http = new Http();
      HttpResponse res = http.send(req);

      if (res.getStatusCode() == 200) {
          // Process the response
```

```
        } else {
            // Log the error
            System.debug('Error ' + res.getStatus());
        }
    } catch (Exception e) {
        System.debug('Exception ' + e.getMessage());
    }
}
```

2. Retry Mechanism

Implement retry logic for failed callouts.

3. Logging Errors

Log errors to a custom object for tracking and monitoring.

16.Explain the concept of Bulk API in Salesforce?

Bulk API is designed to handle large volumes of data for asynchronous processing. It's useful for data migration and batch data integration scenarios.

Use Case Import a large data of accounts into Salesforce. Use tools like Data Loader to upload the CSV file to Bulk API.

17.What are the different types of API protocols used in Salesforce?

REST API: Uses REST principles and supports JSON and XML formats.

SOAP API: Uses SOAP protocol and XML format.

Bulk API: Designed for handling large volumes of data.

Streaming API: For receiving real-time notifications.

Metadata API: For accessing and manipulating Salesforce metadata.

Chatter API: For integrating with Chatter feeds.

Apex REST API: Custom REST API built using Apex.

Apex SOAP API: Custom SOAP API built using Apex

18.What is the importance of HTTP request and response classes in Salesforce integration?

HttpRequest Class: Used to construct HTTP requests (e.g: setting endpoint, method, headers, and body).

HttpResponse Class: Used to handle the response from an HTTP callout (e.g: checking status code, getting response body).

19.What are the best practices for designing integrations in Salesforce?

Best practices for designing integrations in Salesforce include:

- Use Standard APIs: Prefer standard Salesforce APIs (REST, SOAP) for integration.
- Error Handling: Implement robust error handling and retry mechanisms.
- Security: Use OAuth 2.0, Named Credentials, and IP whitelisting for secure integrations.
- Governor Limits: Be mindful of Salesforce governor limits (e.g., API call limits).
- Logging: Implement logging to track integration activities and errors.
- Scalability: Design integrations to handle large data volumes if needed.

20.What is the role of Connected Apps in Salesforce integration?

- Configure OAuth settings for third-party applications.

- Grant specific permissions to apps accessing Salesforce data.
- Track usage and manage connected app access.

21. How do you parse JSON responses in Salesforce Apex?

To parse JSON responses in Salesforce Apex, use the JSON class:

Sample Code

```
public class JsonParser {
    public void parseResponse(String jsonResponse) {
        // Define a wrapper class for JSON structure
        public class Wrapper {
            public String name;
            public String value;
        }

        Wrapper response = (Wrapper) JSON.deserialize(jsonResponse,
Wrapper.class);
        System.debug('Name: ' + response.name);
        System.debug('Value: ' + response.value);
    }
}
```

22. What is an External Data Source, and how is it used in Salesforce Connect?

An External Data Source defines how Salesforce connects to an external system to access data in real-time. It's used in Salesforce Connect to integrate external data without storing it in Salesforce.

23. What are Platform Events?

Platform Events enable event-driven integration by allowing Salesforce to publish and subscribe to events, facilitating real-time data sharing between Salesforce and external systems.

24.How do you handle large data volumes in Salesforce integration?

- Bulk API: Designed for large data volumes and batch processing.
- Data Loader: For manual data import/export.
- Third-Party ETL Tools: Tools like Informatica, MuleSoft for data migration and synchronization

25.What is the importance of IP whitelisting in Salesforce integrations?

IP whitelisting restricts API access to trusted IP ranges, enhancing security by ensuring that only requests from specified IP addresses can access Salesforce data.

26.How do you create a custom REST API in Salesforce using Apex?

Expose the REST API: The class and methods annotated with @RestResource, @HttpGet, and @HttpPost expose the custom REST API endpoints.

Sample Code:

```
@RestResource(urlMapping='/myservice/*')
global with sharing class MyService {
    @HttpGet
    global static String doGet() {
        return 'Hello, World!';
    }

    @HttpPost
    global static String doPost(String input) {
        return 'Received: ' + input;
    }
}
```

27.How do you handle asynchronous processing in Salesforce integrations?

- Future Methods: For long-running operations.
- Queueable Apex: For chaining jobs and more complex processing.
- Batch Apex: For processing large volumes of data in batches.
- Platform Events: For event-driven processing and real-time integration

28.What is Salesforce Metadata API, and how is it used?

Salesforce Metadata API allows you to manage and deploy customizations and configurations programmatically. It's used for retrieving, deploying, creating, and updating metadata components.

29.How do you test HTTP callouts in Salesforce?

When testing Apex code that makes HTTP callouts, you need to use mock callouts to ensure your tests don't actually make external HTTP requests.

1.Implement the HttpCalloutMock Interface:

```apex
public class MockHttpResponse implements HttpCalloutMock {
    public HTTPResponse respond(HTTPRequest req) {
        HttpResponse res = new HttpResponse();
        res.setHeader('Content-Type', 'application/json');
        res.setBody('{"success":true}');
        res.setStatusCode(200);
        return res;
    }
}
```

HttpCalloutMock Interface: This interface requires the respond method, which Salesforce calls when the HTTP callout is made.

Mock Response: The respond method creates an HttpResponse object, sets the headers, body, and status code, and returns it

2.Set Mock Callout in Test:

```
private class CalloutTest {
    @isTest static void testCallout() {
        Test.setMock(HttpCalloutMock.class, new MockHttpResponse());
        // Call the method that performs the callout
    }
}
```

Request and Reply	Fire and Forget	Batch Data Synchronization	Remote Call-In	Data Virtualization
• External Services Using Flow • LWC Calling External System, Trigger • External Services using Batch Apex	• Platform Events using Flow • Platform Events using Apex • Outbound Messages • Apex Based Callouts	• Change Data Capture • Using ETL tool	• SOAP and Rest API • Bulk API • Platform Events • Custom API using Apex,	• Salesforce Connect • Request and reply using LWC • Flow and Apex

30.What is WSDL, and what are the different types in Salesforce?

WSDL (Web Services Description Language) is a document that describes the web service and how to communicate with it. In Salesforce, there are two types of WSDL:

Enterprise WSDL: Specific to a Salesforce org and strongly typed. Changes when the org's metadata changes

Partner WSDL: Loosely typed and designed for use with multiple Salesforce orgs. Does not change with org metadata changes.

31.What is OAuth 2.0? (very important)

OAuth 2.0 is a framework that allows <mark>secure authorization of external applications without sharing user credentials</mark>. It involves exchanging an <mark>authorization code for an access token</mark>, which is then used to access resources.

32.Which OAuth 2.0 flow is used for server-to-server integration without real-time user involvement?

The JWT Bearer Flow is used for server-to-server integration without real-time user involvement.

33.What are Types of Streaming API in Salesforce?

- PushTopic Events
- Generic Events
- Platform Events
- Change Data Capture (CDC)

34.What is Change Data Capture (CDC) in Salesforce?

CDC is a feature that allows real-time tracking and publishing of changes (create, update, delete, undelete) to Salesforce records.

35.What is a Platform Event in Salesforce?

Platform Events are part of the Streaming API and follow the publisher-subscriber model. They are used to define custom payloads and <mark>publish events based on specific criteria.</mark>

Publishing Options:

- Apex
- Flows
- APIs

Subscribing Options:

- Flows
- Triggers
- Lightning Web Components (LWC)

36.What are the REST API HTTP methods available in Salesforce?

GET: Retrieve data.

POST: Create data.

PUT: Update data.

DELETE: Delete data.

PATCH: Partially update data.

37.How many callouts can be made in a single Apex transaction?

In a single Apex transaction, you can make up to 100 callouts to HTTP or web services.

38.What is the difference between Push Technology and Pull Technology in the context of Streaming API?

Push Technology: The server sends updates to the client in real-time when there are changes or events, without the client needing to request (or poll) for updates.

Pull Technology: The client must repeatedly request (or poll) the server for updates, which can be less efficient and slower compared to push technology.

39.What are the main differences between Platform Events and Change Data Capture (CDC) in Salesforce?

Platform Events:

- Custom-defined payloads.
- Can be published based on specific criteria without performing DML operations.
- Suffix with _e in API name.

Change Data Capture (CDC):

- Automatically publishes events when records are created, updated, deleted, or undeleted.
- Publishes only the changed data.
- Requires enabling the setting for the specific objects.

40.What is the main use case for the Data Virtualization pattern in Salesforce?

The main use case for the Data Virtualization pattern is to access and use data from external systems in Salesforce without physically storing it within Salesforce. This can be achieved using Salesforce Connect.

41.When would you use the Remote Call-In pattern in Salesforce?

The Remote Call-In pattern is used when an external system needs to make API calls to Salesforce.

42.What is the difference between Enterprise WSDL and Partner WSDL in Salesforce?

Enterprise WSDL:

- Strongly typed.
- Specific to a Salesforce organization.

- Changes if custom fields or objects are added or modified in the Salesforce org.
- Best for use within a single Salesforce organization.

Partner WSDL:

- Loosely typed.
- Generic and can be used across multiple Salesforce organizations.
- Does not change with modifications to custom fields or objects.
- Best for use with applications that need to integrate with multiple Salesforce orgs.

43.What are the main actors involved in OAuth 2.0 and their roles?

Resource Owner: The user who authorizes an application to access their account.

Client: The application requesting access to the resource server on behalf of the resource owner.

Resource Server: The server hosting the protected resources (e.g., Salesforce).

Authorization Server: The server issuing access tokens to the client after successfully authenticating the resource owner.

44.When should you use JWT Bearer Flow in OAuth 2.0 for Salesforce?

JWT Bearer Flow is ideal for server-to-server integrations without real-time user involvement. It is used when an application needs to access Salesforce via API calls, such as in ETL tools or middleware systems, where no user interaction is required.

45.What is the range of successful integration response codes in Salesforce?

The range of successful integration response codes is typically from 200 to 299. These include:

200 (OK): The request was successful.

201 (Created): The request was successful and a new resource was created.

202 (Accepted): The request has been accepted for processing, but the processing is not complete.

46.Why is the POST method used for getting an access token in Salesforce? (important)

The POST method is used for getting an access token in Salesforce because it involves generating and creating a new unique token for the user session. This token is then stored and used for subsequent authentication without needing to resend user credentials.

47.How do you handle JSON responses in Salesforce Apex? (important)

JSON responses can be handled using the following methods:

JSON.deserialize(): Converts a JSON string into a strongly-typed Apex object.

JSON.deserializeUntyped(): Converts a JSON string into a generic Apex object (usually a Map or List).

48.What are the common status codes received from HTTP responses, and what do they indicate?

200 (OK): The request was successful.

201 (Created): The request was successful, and a new resource was created.

400 (Bad Request): The request was malformed or invalid.

401 (Unauthorized): Authentication is required or failed.

403 (Forbidden): The server understands the request but refuses to authorize it.

404 (Not Found): The requested resource could not be found.

500 (Internal Server Error): The server encountered an error while processing the request.

49.What is the benefit of using Wrapper Classes for handling JSON in Salesforce? (important)

Wrapper Classes provide a structured and strongly-typed way to handle JSON data. They simplify the process of parsing JSON responses and make the code more readable and maintainable.

50.What are the key differences between Salesforce REST API and APEX REST API?

Salesforce REST API allows updating records without writing code. It uses standard Salesforce APIs that are updated with each release.

APEX REST API customizable API that allows creating custom methods and updating multiple records in a single transaction.

51.How do you construct a REST API call URL in Salesforce?

- Salesforce instance URL

- API type (usually /services/data/)
- API version (e.g., v47.0)
- Resource (e.g., sobjects/Account/)
- Record ID (optional, for specific record operations)

52. How do you insert a record using REST API in Salesforce?

- Construct the endpoint URL (e.g: https://instance.salesforce.com/services/data/v47.0/sobjects/Case/).
- Use the POST method.
- Include the record data in JSON format in the request body.
- Send the request using tools like Workbench or Postman.
- Parse the response to get the record ID and status.

53. How do you query records using REST API in Salesforce?

- Construct the endpoint URL (e.g: https://instance.salesforce.com/services/data/v47.0/query/?q=SELECT+Id,+Name+FROM+Account).
- Use the GET method.
- Send the request using tools like Workbench or Postman.
- Parse the response to retrieve the queried data.

54. How do you handle JSON data in Salesforce REST API calls? (important)

Use the JSON.serialize() method to convert Apex objects to JSON format for the request body.

Use the JSON.deserialize() or JSON.deserializeUntyped() methods to parse JSON responses into Apex objects or maps.

55. What is the Content-Type header used for in REST API calls?

The Content-Type header specifies the media type of the request body. For Salesforce REST API calls, it is typically set to application/json to indicate that the request body contains JSON data.

56.What is the difference between PUT and PATCH methods in REST API? <mark>(important)</mark>

PUT: Used to update or insert a resource. It replaces the entire resource with the data provided.

PATCH: Used to update partial resources. It only updates the fields provided in the request body without affecting other fields.

57.How do you handle authentication for REST API calls in Salesforce?

- Create a connected app in Salesforce to obtain client ID and client secret.
- Send a POST request to the OAuth 2.0 token endpoint with the client ID, client secret, username, password, and grant type to obtain an access token.
- Use the access token in the Authorization header for subsequent API requests.

58.What is the purpose of the @RestResource annotation in an APEX class?

The @RestResource annotation is used to expose an Apex class as a RESTful web service.

59.What are some common tools used for testing and interacting with REST APIs in Salesforce?

- Workbench
- Postman
- Curl
- Salesforce Developer Console

60.How do you retrieve all cases created today using Salesforce REST API?

- Construct the SOQL query: SELECT Id, Subject FROM Case WHERE CreatedDate = TODAY.
- Construct the endpoint URL (https://instance.salesforce.com/services/data/v47.0/query?q=SELECT+Id,+Subject+FROM+Case+WHERE+CreatedDate=TODAY).
- Use the GET method.
- Send the request using tools like Workbench or Postman.
- Parse the response to retrieve the queried data.

61.What is the "Fire and Forget" integration pattern, and when should it be used?

The "Fire and Forget" integration pattern is used when Salesforce sends a message to an external system without waiting for a response. The best practice for this pattern is to use Platform Events.

62.What is the role of scopes in OAuth?

Scopes define the specific permissions that a client application is requesting from the resource owner. They specify what type of access the application needs, such as reading or writing data, and help limit the permissions granted to the client.

63.What is the difference between the web server flow and the user-agent flow in OAuth? (very important)

The web server flow is used for web applications that can securely store the <mark>client secret on the server side.</mark> In contrast, the user-agent flow is used for applications like mobile apps where the <mark>client secret cannot be securely stored</mark> because the application is running on the user's device.

64.What is the significance of the callback URL in OAuth? <mark>(important)</mark>

The callback URL is the URL to which the <mark>authorization server redirects the resource owner after successful authentication</mark>.

65.How does the refresh token work in OAuth?

The refresh token is a long-lived token used to obtain new access tokens without requiring the resource owner to re-authenticate. It helps maintain continuous access by allowing the client application to refresh the access token periodically.

66.When should you use JWT Bearer Token Flow instead of OAuth Web Server Flow?

Use JWT Bearer Token Flow for server-to-server integration scenarios where <mark>no user agent involvement is required</mark>. Use OAuth Web Server Flow when user interaction is necessary for authorization.

67.What components make up a JWT (JSON Web Token)?

A JWT consists of three components: <mark>Header, Payload, and Signature.</mark>

68.What role does the certificate play in JWT Bearer Token Flow?

The certificate is used to sign the JWT request, authenticate the connected app, and obtain the access token.

69.What can I do with an External Object?

	Standard and Custom Objects	External Objects
Write	X	
Read	X	X
REST and SOAP API	X	X (read only)
Custom Tabs and Details	X	X (read only)
SOQL	X	X (read only, no aggregation, no tracking)
Search and SOSL	X	X
Reports and Analytics	X	

70.We're designing a system for a client that involves quite a bit of data. The regular data storage limits in Salesforce will be exceeded by a factor of 10 or more. Which technology is recommended?

One technology that seems like it might be cost effective is to store large static sections of data virually and access it using External Objects

71.Explain Event-Driven software architecture?
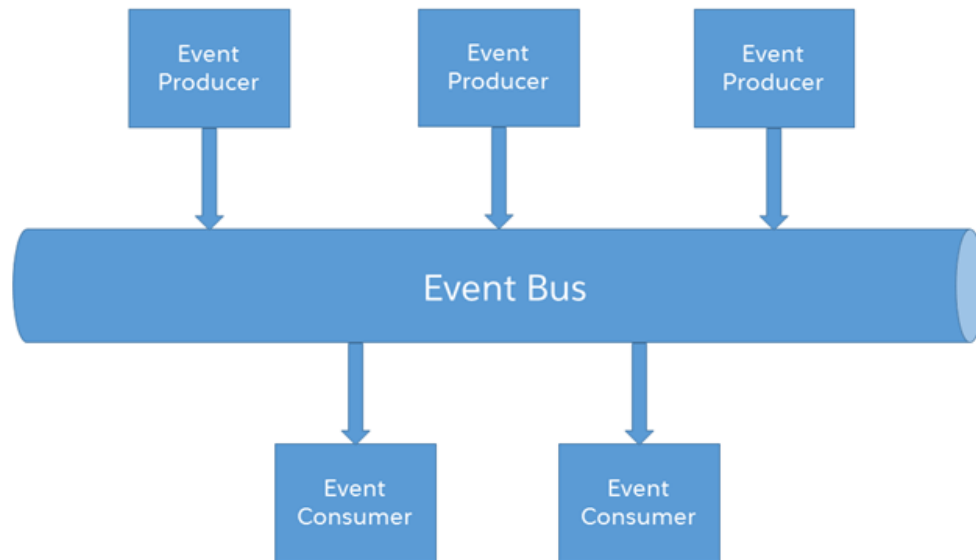
Event

A change in state that is meaningful in a business process.

Event message

A *message that contains data about the event*. Also known as an event notification.

Event producer

The publisher of an event message.

Event channel

A stream of events on which an event producer sends event messages and event consumers read those messages.

Event consumer

A subscriber to a channel that receives messages from the channel.

Event bus

The event bus enables the retrieval of stored event messages at any time during the retention window.

Explain The role of Event Bus in platform event?

Platform event messages are published to the event bus, where they are **stored temporarily**.

72.What are considerations for Platform Events ?

- Unlike custom or standard objects, you can't update or delete event records.

- You also can't view event records in the Salesforce user interface.

- Platform events don't have page layouts.

- When you delete a platform event definition, it's permanently deleted.

- You can set read and create permissions for platform events. Grant permissions to users in profiles or in permission sets.

73.Explain Publish Behavior of platform event?

- **Publish After Commit** to have the event message published only after a transaction commits successfully.

- **Publish Immediately** to have the event message published when the publish call executes. Select this option if you want the event message to be published regardless of whether the transaction succeeds.

74.Can apex trigger on platform events?

Yes

75.What is WSDL? (important)

A WSDL is an **XML Document** which contains a standardized description of how to **communicate using webservice**.

76.What is difference between SOAP and Bulk API?

SOAP is synchronous and BULK is asynchronous.

Synchronous means that the batches are updated one at a time, in order of when the job was queued and you have to finish one before you can start the next. Asynchronous can run in parallel and doesn't require previous entries to finish before starting the next batch.

77.When to Use Bulk API?

Bulk API is based on REST principles and is optimized for loading or deleting large sets of data. You can use it to query, queryAll, insert, update, upsert, or delete many records asynchronously by submitting batches. Salesforce processes batches in the background.

The easiest way to use Bulk API is to enable it for processing records in Data Loader using CSV files.

78.Can we make callout directly from trigger?

No we cannot. Callout is a Asynchronous process where as Trigger is Synchronous. Callouts would hold up the database transaction until the callout completed. That means it is not directly possible to do a web service callout from a trigger. But using @Future annotation we can convert the Trigger into a Asynchronous and we can use a Callout method.

79.How to use HTTP Callouts in batch class? (important)

To use HTTP Callouts in batch class we need to use Database.allowcallouts in interface.

80.What is OpenID Connect?

OpenID Connect is a <mark>simple identity layer on top of the OAuth 2.0 protocol.</mark> It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User.

81.How do you make a callout in Salesforce?

Http http = new Http();

HttpRequest request = new HttpRequest();

request.setEndpoint('https://api.example.com/data');

request.setMethod('GET');

HttpResponse response = http.send(request);

if (response.getStatusCode() == 200) {

   // Process the response

}

82.How does MuleSoft enhance Salesforce integrations? (important)

MuleSoft provides an integration platform that connects Salesforce to various systems using <mark>pre-built connectors</mark> and integration templates. It handles data mapping, transformation, and orchestration, simplifying complex integrations.

83.When to use Chatter REST API?

Chatter REST API is used to display chatter feeds, users, groups, and followers, especially in mobile applications. It provides programmatic access to files, recommendations, topics, notifications, and more.

84.When to Generate WSDL in SOAP API Integration?

When You Need to Generate WSDL:

When you want to expose Salesforce functionality to an external system. You create an <mark>Apex class with web service methods</mark> that external systems can invoke. Generate a WSDL file for your Apex class.

The WSDL file is shared with the external system so they can understand and consume the web services provided by Salesforce.

Example: An external ERP system needs to access Salesforce data or functionality.

My Name is Smriti Sharan. I am [avid blogger](#) and [youtuber](#). Follow my Blog and youtube to learn various aspect of Salesforce.

https://t.me/sfdcamplified feel free to join telegram group.