

what is HTTP, SOAP, REST, ?

Adapters: JDBC, SAP, REST, JMS

Languages: Flow, Java, SQL

Protocols: HTTP, SOAP, REST, FTP, JMS

Database: Oracle, MySQL, SQL Server

Operating Systems: Windows, Linux

Version Control: Git, SVN

1.) what is integration in webmethods, interviewer asked me that question in interview, pls answer to it in a small definition.

Ans.) Integration in webMethods involves connecting different software applications, systems, and data sources to enable seamless communication and data exchange. It uses components like the Integration Server, adapters, and APIs to ensure that disparate systems work together efficiently, supporting real-time processing, data transformation, and secure data flow across an enterprise. It involves Integration Server, Universal Messaging, API Gateway, Adapters, webMethods Designer, Flow Services, Business Process Management, API Management, B2B Integration

In summary, integration in webMethods involves leveraging its powerful platform to connect, manage, and automate interactions between various systems and applications, thereby streamlining business operations and ensuring efficient data flow across the enterprise.

2.) what is HTTP?

Ans.) HTTP, or HyperText Transfer Protocol, is a communication protocol used on the internet to transfer data between a web server and a web browser. It defines how messages are formatted and transmitted, and how web servers and browsers should respond to various commands. HTTP is the foundation of any data exchange on the Web and facilitates the loading of web pages.

3.)What is FTP?

Ans.)FTP, or File Transfer Protocol, is a standard network protocol used to transfer files between a client and a server over the internet or a local network. It allows users to upload, download, and manage files on a remote server.

4.)what is SOAP?

Ans.)SOAP (Simple Object Access Protocol) is a protocol used for exchanging structured information in the implementation of web services. It relies on XML to format the data and usually operates over HTTP or SMTP. SOAP allows programs running on different operating systems to communicate with each other by using standard protocols.

5.)what is REST?

Ans.)REST, or Representational State Transfer, is an architectural style for designing networked applications. It uses standard HTTP methods (such as GET, POST, PUT, DELETE) to enable communication between a client and a server. RESTful services are stateless, meaning each request from a client to a server must contain all the information needed to understand and process the request. REST is widely used for developing APIs due to its simplicity, scalability, and compatibility with web standards.

Adapters:

6.)what is JDBC adapter?

Ans.) A JDBC (Java Database Connectivity) adapter is a software component that allows a Java application to interact with a relational database. It provides an interface for connecting to a database, executing SQL queries, and retrieving results, making it easier for developers to perform database operations from within a Java program.

7.)What is SAP adapter?

Ans.) An SAP adapter is a middleware component that enables communication between SAP systems and other software applications. It facilitates the integration of SAP applications with external systems by translating data and requests into formats that both the SAP system and the external applications can understand. This allows seamless data exchange and process automation across different platforms.

8.) what is KAFKA adapter?

Ans.) A Kafka adapter is a software component that enables the integration of Apache Kafka with other systems or applications. It allows data to be published to and consumed from Kafka topics, facilitating real-time data streaming and processing. Kafka adapters are often used to connect Kafka with databases, messaging systems, data processing frameworks, and other data sources or sinks, ensuring smooth data flow and interoperability across different systems.

Real Time questions:

9.) what is flow service in webmethods?

Ans.) In webMethods, a Flow service is a graphical programming construct used to design and implement integration logic. It allows developers to create workflows by connecting predefined services and custom logic using a drag-and-drop interface. Flow services are used to automate business processes, perform data transformations, and integrate different systems and applications within an enterprise. They are a key component of the webMethods Integration Server, enabling efficient and visual development of complex integration scenarios.

10.) what is synchronous interface?

Ans.) In webMethods, a synchronous interface refers to a method of communication where a calling application or service sends a request to another service and waits for an immediate response before continuing its execution. This means that the calling process is blocked during the operation, ensuring that tasks are performed in a sequential and predictable manner.

Use Cases: Synchronous interfaces are commonly used in real-time systems where immediate feedback is necessary, such as:

- a.) Online transactions
- b.) User authentication
- c.) Real-time data retrieval
- d.) Interactive user interfaces where users expect immediate responses.

Implementation in webMethods: In webMethods, synchronous interfaces can be implemented using various components such as:

Web Services: Both SOAP and RESTful web services can be designed to operate synchronously, where a client waits for the service response before proceeding.

Remote Procedure Calls (RPCs): Methods invoked on remote systems that require immediate feedback to the caller.

HTTP Requests: Direct HTTP requests to webMethods Integration Server that follow a synchronous pattern.

In summary, a synchronous interface in webMethods is characterized by its immediate response requirement and blocking nature, making it suitable for scenarios where real-time interaction and immediate data processing are crucial.

11.)what is asynchronous interface?

Ans.) In webMethods, an asynchronous interface is a method of communication where the calling application or service sends a request and continues its execution without waiting for an immediate response. The response or result of the request is handled separately, often through callbacks, events, or message queues. This approach allows for non-blocking operations, enabling the calling process to perform other tasks concurrently.

Implementation in webMethods:

Message Queues: Using JMS (Java Message Service) or webMethods Messaging, messages can be published to a queue or topic, and consumers can process these messages asynchronously.

Callbacks: Implementing callbacks where the calling service provides a callback function to be invoked once the response is ready.

Event Listeners: Setting up event listeners that trigger specific actions when certain events occur.

Use Cases:

Data Integration: Integrating with external systems where the response time is uncertain or variable.

Background Processing: Performing tasks like data aggregation, report generation, or cleanup operations in the background.

User Notifications: Sending email or SMS notifications where the sending process is decoupled from the main application logic.

12.)what are the different methods to invoke the service?

Ans.) a.) Invoke step within webmethods designer.

b.)Transformers in map step.

c.)using remote invoke built in service can invoke built in flowservice in webmethods environment.

d.)using webmethods supported protocols like http and https

d.)using file polling in webmethods

13.)how to invoke a service from the browser?

Ans.) To invoke a service from a browser in webMethods, you typically use an HTTP request to interact with webMethods Integration Server services.

Eg: `http://<host>:<port>/invoke/<package_name>/<folder_name>/<service_name>`

14.) when you create and save the flow "my.pack:myflow" In the "MyPack" package where will you find the code?

Ans.) In the `MyPack\src\my\pack\myFlow\flow.xml` file it is saved in the form of flow.xml file and the entire code will be saved in the form of xml file.

15.)what is implicit and explicit mapping in webmethods , give small definition?

Ans.)Implicit Mapping:

Implicit mapping automatically maps variables with the same names and compatible data types between the source and target. This reduces the need for manual mapping but requires that the variable names in both structures match exactly.

Explicit Mapping:

Explicit mapping involves manually specifying the mapping between source and target variables. This allows for greater control and flexibility, enabling you to map variables with different names and apply transformations or custom logic during the mapping process.

16.)what is schema in webmethods

Ans.)In webMethods, a schema defines the structure and data types of the information that a service or application processes. It serves as a blueprint for the data, ensuring that it conforms to a predefined format, which is crucial for data validation, transformation, and integration. Here are some key points about schemas in webMethods:Types of Schemas

XML Schema (XSD):

An XML Schema Definition (XSD) is used to define the structure of XML documents. It specifies the elements, attributes, and data types allowed in the XML document.

In webMethods, XSDs are often used to validate XML data before processing it in a service.

Flat File Schema:

A Flat File Schema defines the structure of flat file data, such as CSV or fixed-width files.

It specifies how the data is organized, including delimiters, record definitions, and field definitions.

JSON Schema:

A JSON Schema defines the structure of JSON documents, specifying the data types, required fields, and allowed values.

It is used for validating JSON data in webMethods services.

17.)what is the service you will use in error sequence to in error details ?

Ans.) In webMethods, the service typically used in an error sequence to log or handle error details is `pub.flow:getLastError`. This service is designed to retrieve the last error that occurred in the current context, providing detailed information about the error.

Using `pub.flow:getLastError`

Purpose:

The `pub.flow:getLastError` service is used to obtain details about the most recent error, including the error message, error code, and the call stack.

Typical Usage in an Error Handling Sequence:

Invoke `pub.flow:getLastError`:

This service is invoked to capture the error details whenever an error occurs in the flow.

Process Error Details:

The error details returned by `pub.flow:getLastError` can be used to log the error, send notifications, or perform specific error handling logic.

18.)what are the primary purpose of web service connector?

Ans.) A Web Service Connector (WSC) in webMethods enables communication with external web services. It facilitates service invocation, allowing webMethods Integration Server to send requests and receive responses from SOAP or RESTful services. WSCs ensure interoperability using standard protocols and data formats, simplifying integration through automated client stub generation. They support secure communication and authentication, handling errors and retries robustly. WSCs can be created using a wizard in webMethods Designer, making it easier to connect and interact with external services for various integration scenarios.

19.)when coding in services, how can a document type be represented in java?

Ans.) In webMethods, when coding services in Java, a document type is represented by the `IData` interface. `IData` is the primary interface for manipulating webMethods document types in Java services. Here's how you can represent and work with a document type in Java.

20.)what are the different steps in deployer to deploy the code.

Ans.) a.) create the deployer project

b.)Define step - select the packages

c.)Build step - create the build

d.)map - select the target servers

e.)deployment candidate - to do simulation, checkpoint and deployment

21.) can u explain get lost error in webmethods?

Ans.) It catches the error in get last error.

b.) log the error details (service name, date time of the error and exception details and other key information in a separate line. Log the error details using debug log.

22.) how to get the current index of the list in a loop?

Ans.) There is a special variable on the pipeline called \$iteration which will be incremented as the loop operator passes through the list.

23.) what is WSDL? what does a WSDL file contains ? Give information in short form

Ans.) WSDL (Web Services Description Language) is an XML-based language used to describe web services, specifying their functionality, how to access them, and the communication protocols and data formats used.

It contains request Structure, response structure URL endpoint details required for SOAP webservice implementation.

24.) How will u create SOAP based webservice in webmethods?

Ans.) 1.) New - webservice descriptor

2.) select the description -- provider/consumer

3.)if it is provider, select the list of service of which we want to choose and click finish(each service will be exposed as a operation and download the WSDL and starting it from external app.)

4.)if it is consumer then browse the WSDL details then consume and then click finish.

25.)how to create REST service in webmethods?

Ans.) 1.) New - REST Resource

2.) select the resource type

3.) and select the the checkbox for generate JSON API URL templates and then click finish.

26.) Give the difference between REST and SOAP webservice in short form

Ans.) REST - Representational State Transfer Protocol

SOAP - Simple object Access Protocol

Difference Between REST and SOAP Web Services

a.)Protocol:

SOAP: Uses strictly defined protocols like HTTP, SMTP, and XML-RPC.

REST: Uses HTTP protocols and is more flexible, often employing JSON or XML.

b.)Message Format:

SOAP: Uses XML for message format.

REST: Can use multiple formats, commonly JSON or XML.

c.)Complexity:

SOAP: More complex with rigid standards and extensive protocols.

REST: Simpler, uses standard HTTP methods like GET, POST, PUT, DELETE.

d.)Statefulness:

SOAP: Can be stateful with built-in support for complex transactions.

REST: Stateless, each request from a client to server must contain all the information needed to understand and process the request.

e.)Performance:

SOAP: Generally slower due to XML parsing and the complexity of standards.

REST: Faster due to lightweight payloads, typically using JSON.

27.)what is EAI/ B2B?

Ans.) a.) EAI (Enterprise Application Integration) is the process of linking different enterprise applications within an organization to simplify and automate business processes, without making significant changes to the existing applications or data structures.To enable seamless communication and data exchange between disparate systems. Includes middleware, adapters, and messaging systems to connect applications.

b.) B2B (Business-to-Business) refers to commerce transactions between businesses, typically involving manufacturers, wholesalers, distributors, or other companies. It involves the exchange of goods, services, or information between businesses rather than between businesses and consumers (B2C).Facilitates trade and transactions between companies, often in bulk quantities.Include procurement, supply chain management, and partnership agreements.

28.) what are the adapters used in webmethods?

Ans.) a.) JDBC adapter : A JDBC (Java Database Connectivity) adapter is a software component that enables connectivity between Java applications and relational databases. It allows Java programs to execute SQL queries, retrieve data, and manipulate databases through a standardized interface.

b.) MQ adapter : An MQ (Message Queue) adapter facilitates communication between applications by enabling them to send and receive messages asynchronously via message queues. It ensures reliable and orderly message delivery across distributed systems.

c.) KAFKA adapter : It is a very big data that we store the data in this database. A Kafka adapter integrates applications with Apache Kafka, a distributed streaming platform. It enables publishing and subscribing to streams of records, facilitating real-time data processing and communication between systems.

And many more adapters are there to store the data and we used in webmethods.

29.) Tell me the different types of templates in jdbc adapter

Ans.) The JDBC adapter in webMethods provides various templates to simplify database operations. These templates are categorized based on the type of database interaction they perform. Here are the different types of templates:

1. Basic Templates

Insert: Inserts a new record into a database table.

Update: Updates existing records in a database table.

Delete: Deletes records from a database table.

Select: Retrieves records from a database table based on specified criteria.

2. Advanced Templates

Stored Procedure: Executes a stored procedure in the database.

Batch Insert: Inserts multiple records into a database table in a single batch operation.

Batch Update: Updates multiple records in a database table in a single batch operation.

Batch Delete: Deletes multiple records from a database table in a single batch operation.

Dynamic SQL: Executes dynamically constructed SQL statements at runtime.

3. Custom SQL Templates

CustomSQL: Allows you to write and execute custom SQL queries that do not fit into the standard operations provided by the other templates.

Summary

Basic Templates: Insert, Update, Delete, Select.

Advanced Templates: Stored Procedure, Batch Insert, Batch Update, Batch Delete, Dynamic SQL.

Custom SQL Template: CustomSQL.

30.) what is finally block in webmethods explain?

Ans.) In webMethods, the finally block is used within a SEQUENCE step to ensure that certain actions are performed regardless of whether an error occurred in the preceding steps. It is similar to the finally block in traditional programming languages like Java.

Key Points:

Usage Context:

It is typically used within a try-catch-finally construct, where the try block contains the main logic, the catch block handles any errors, and the finally block contains cleanup code or final actions.

Execution:

The finally block is executed after the try and catch blocks, regardless of whether an exception was thrown or not.

This makes it ideal for tasks such as closing resources, logging, or any other cleanup activities that need to be performed no matter what.

SEQUENCE (Exit on SUCCESS)

SEQUENCE (Exit on FAILURE) [TRY Block]

- Step1: Some service call that may fail
- Step2: Another service call

SEQUENCE (Exit on DONE) [CATCH Block]

- getLastError: pub.flow:getLastError
- logError: Log the error details

SEQUENCE (Exit on SUCCESS) [FINALLY Block]

- cleanup: Perform cleanup tasks (e.g., close connections, log completion)

The TRY block contains service calls that might fail.

The CATCH block retrieves error information and logs it.

The FINALLY block performs cleanup tasks, such as closing resources or logging the completion of the process.

31.) what is load balancing in webmethods?

Ans.) Load balancing in webMethods Integration Server (IS) ensures that incoming requests are distributed across multiple server instances to optimize resource use, improve response times, and increase reliability.

Key Points:

Clustering:

Definition: A group of Integration Servers working together as a single unit to distribute workload.

Configuration: Set up a cluster by configuring multiple IS instances to work together.

Load Balancer:

Purpose: Acts as an intermediary that distributes incoming client requests across the available IS instances.

Types: Can be hardware-based (like F5, Cisco) or software-based (like Apache HTTP Server with mod_proxy, Nginx).

32.) what are the different types of testing in webmethods have?

Ans.) In webMethods, different types of testing ensure the reliability, performance, and functionality of the integrations and services. Here are the primary types of testing used in webMethods:

1. Unit Testing

Definition: Testing individual components or services to ensure they function as expected.

Tools: webMethods Unit Test Framework, custom test harnesses.

Purpose: Validate the smallest parts of the application independently.

2. Integration Testing

Definition: Testing the interaction between different modules or services to ensure they work together correctly.

Tools: webMethods Integration Server, Integration Testing tools (e.g., SOAP UI, Postman).

Purpose: Verify that integrated components or systems function together as intended.

3.) User Acceptance Testing (UAT):

Definition: Testing conducted by the end-users or clients to ensure the system meets their requirements and is ready for production.

Tools: End-user specific testing environments and tools.

Purpose: Validate the solution from the end-user perspective before going live.

4.) Performance Testing

Definition: Testing to determine how a system performs in terms of responsiveness and stability under a particular workload.

Tools: Load testing tools (e.g., JMeter, LoadRunner).

Purpose: Identify bottlenecks, ensure the system can handle expected load, and maintain performance standards.

33.)what is the use of clear pipeline in the webmethods?

Ans.) The clearPipeline service in webMethods Integration Server is used to remove variables from the pipeline. The purpose of using clearPipeline is to manage and control the data in the pipeline, ensuring that unnecessary or sensitive data does not persist and affect subsequent services or steps in the flow.

The clearPipeline service is essential for managing the pipeline effectively in webMethods Integration Server. It ensures efficient memory usage, enhances security by removing sensitive data, maintains data integrity, and simplifies debugging and maintenance of integration services.

34.) what are the other companies which provides integration tool?

Ans.) MuleSoft, software AG webmethods, Dell Bhoomi and Tibco etc.

35.)what are the minimum and maximum pool in webmethods JDBC adapter?

Ans.)In webMethods JDBC adapter, the minimum and maximum pool settings define the number of database connections that the adapter maintains in its connection pool. These settings are critical for optimizing database connectivity and performance.

Key Points:

Minimum Pool Size:

Definition: The minimum number of connections that the pool maintains at all times.

Purpose: Ensures that a certain number of connections are always available, reducing the overhead of creating connections on demand.

Typical Setting: Depends on the expected load and resource availability. A common starting point might be 1-5 connections.

Maximum Pool Size:

Definition: The maximum number of connections that the pool can maintain.

Purpose: Limits the number of concurrent connections to the database to prevent overloading the database server and consuming too many resources.

Typical Setting: Depends on the maximum expected load and the database's capacity. A common setting might range from 10 to 50 connections or more, depending on the system's requirements.

Configuration Example:

When configuring a JDBC connection pool in webMethods, you specify these values in the JDBC adapter connection settings:

Minimum Pool Size: 5

Maximum Pool Size: 20

36.)what are webmethods trading network , explain..

Ans.) WebMethods Trading Networks (TN) is a component of the Software AG webMethods suite designed for business-to-business (B2B) integration. It facilitates secure and efficient exchange of electronic documents like EDI, XML, and flat files with trading partners. Key features include partner management, document management, security through encryption and secure protocols, seamless integration with other systems, comprehensive monitoring and reporting, and scalability for handling large transaction volumes. TN automates document exchange, reduces manual effort, ensures data accuracy, and provides robust security. It also helps businesses comply with industry standards and regulations by offering detailed audit trails and reports.

Trading network is also the base through which webmethods products support numerous ebusiness standards Rosattanet, EDI, ebXMLMessaging Service, SWIFT, FIX and EIDX.

37.)where Trading network is installed in webmethods?

Ans.) WebMethods Trading Networks is installed within the Integration Server (IS) environment of the webMethods suite. For example, you can access Trading Networks via the Integration Server's administration console or the My webMethods user interface. In WM Tn package it is installed built in.

38.) Explain about flow steps in web methods?

Ans.)

In webMethods, flow steps are the basic building blocks used to create services in the Flow language. Here are some of the common flow steps:

1. MAP

Purpose: Used to map data from one variable to another, transform data, and perform data manipulations.

Details: You can perform simple assignments, invoke built-in functions (e.g., string manipulations), and call custom Java services within a MAP step.

Usage: Essential for data transformation and preparation before invoking other services or sending data to an external system.

Example: Mapping an input field to an output field, converting string cases, or performing arithmetic operations.

BRANCH:

Purpose: Implements conditional logic to execute different sets of flow steps based on the evaluation of a variable or expression.

Details: Supports pattern matching against variable values or expressions and directs the flow to the corresponding child step.

Usage: Essential for decision-making processes, such as routing logic based on the type of input data.

Example: Branching based on a customer type to execute different business rules for retail and wholesale customers.

SEQUENCE:

Purpose: Groups a set of flow steps and controls their execution as a unit.

Details: Can be configured with different exit conditions such as "SUCCESS", "FAILURE", or "DONE" to manage error handling and execution flow.

Usage: Used for structuring the flow logically, managing error handling, and ensuring certain steps are executed together.

Example: Grouping steps for a transaction, where all steps must succeed for the transaction to be considered successful.

REPEAT:

Purpose: Repeats a set of flow steps until a specified condition is met or a maximum number of iterations is reached.

Details: Allows specifying retry intervals and conditions to manage retries and handle transient errors.

Usage: Useful for retry logic, handling temporary failures, and performing repetitive tasks.

Example: Repeating a service call until a successful response is received or a timeout occurs.

LOOP:

Purpose:

The LOOP step is designed to handle repetitive operations on a list of items, such as processing each record in a batch or iterating over an array of data elements.

Input Array:

you specify the array variable that the LOOP step will iterate over. This variable should be a list (e.g., an array of strings, documents, etc.).

Output Array:

You can define an output array where the results of each iteration can be collected. This is useful for aggregating the processed data.

Index Variable:

An optional index variable can be specified to keep track of the current iteration index. This is useful if you need to reference the position of the current element within the array.

Usage:

Configuring a LOOP Step:

Add a LOOP Step:

In your flow service, add a LOOP step where you need to iterate over a list of elements.

Specify Input Array:

Set the Input Array property to the name of the list variable you want to loop through.

Specify Output Array:

Optionally, set the Output Array property to collect the results of each iteration.

Add Child Steps:

Inside the LOOP step, add the flow steps that you want to execute for each element in the input array.

Summary:

The LOOP step in webMethods is used to iterate over an array, processing each element individually.

Input Array: Specifies the list to iterate over.

Output Array: Optionally collects results of each iteration.

Index Variable: Optionally tracks the current iteration index.

Inside the LOOP, you can add child steps to perform operations on each element, such as data transformations and service invocations.

TRY-CATCH:

In webMethods, a try-catch block is implemented using the SEQUENCE flow steps to handle exceptions and ensure proper error management. The try block (outer SEQUENCE) is set to "exit on SUCCESS" and contains the steps that might throw an exception. The catch block (inner SEQUENCE) is set to "exit on FAILURE" and is nested within the try block to handle any exceptions that occur. When an error is encountered in the try block, control is transferred to the catch block, where you can perform error handling actions, such as logging the error or cleaning up resources. This structure ensures that exceptions are caught and managed gracefully, preventing them from causing unexpected failures in the flow service.

TRY-FINALLY:

In webMethods, a try-finally block ensures certain actions are always performed, regardless of whether an exception occurs. The try block is implemented using an outer SEQUENCE step set to "exit on SUCCESS," containing steps that might throw an exception. A nested SEQUENCE step within the try block, set to "exit on DONE," serves as the finally block and includes cleanup or final actions. This structure guarantees that the finally block executes no matter what happens in the try block, ensuring resource cleanup and consistent state. Here's an example:

xml

Copy code

<flow>

```
<sequence exitOn="SUCCESS"> <!-- Try block -->
```

```
<!-- Steps that might throw an exception -->
```

```
<sequence exitOn="DONE"> <!-- Finally block -->
```

```
<!-- Cleanup or final steps -->
```

```
</sequence>
```


