26 May 2025 11:17

WebMethods -

webMethods is an integration platform developed by Software AG that connects applications, systems, and data across an enterprise. It enables seamless business process integration, API management, and data exchange through its tools like the Integration Server, API Gateway, and Business Process Management. webMethods supports real-time monitoring, scalability, and automation, making it ideal for streamlining workflows, synchronizing data, and ensuring compliance. It is widely used for hybrid integrations, connecting on-premise and cloud-based applications efficiently.

Default port –

Integration Server: 5555

- **MySQL: 3306**
- **Oracle: 1521**
- **MongoDB: 27017**
- **React: 3000**

MAP –

- **The MAP step is used to map input data to output data. It is commonly used for transforming or copying data from one variable to another. For example, if you have a user object and want to map its attributes (first Name, last Name) to another object customer, you use the MAP step.**
- **It is a service that allows you to adjust the contents and structure of a pipeline. It is used to transform name, value, and structure.**
- **You can also use the MAP step for transformations using Built-In Services like string:toLowerCase or string:concat.**
- **The MAP step is flexible and essential in transforming data between different formats such as XML, JSON, flat files, or internal business data.**

Branch –

- **The BRANCH step allows you to conditionally execute a step based on the value of a variable at run time. For example, you might use a BRANCH step to process a purchase order one way if the Payment Type value is "CREDIT CARD" and another way if it is "CORP ACCT".**
- **When you build a BRANCH step, you can:**
- **Branch on a switch value: Use a variable to determine which child step executes. At run time, the BRANCH step matches the value of the switch variable to the Label property of each of its targets. It executes the child step whose label matches the value of the switch.**
- **Branch on an expression: Use an expression to determine which child step executes. At run time, the BRANCH step evaluates the expression in the Label property of each child step. It executes the first child step whose expression evaluates to "true."**

Repeat –

- **The REPEAT step allows you to conditionally repeat a sequence of child steps based on the success or failure of those steps.**
- **The REPEAT step is used to repeat a block of flow steps until a condition is met.**
- **Unlike LOOP, the REPEAT step checks the condition after the block has been executed, so the condition is evaluated at the end of each iteration.**
- **You can use REPEAT to:**
- **Re-execute (retry) a set of steps if any step within the set fails.**
- **Re-execute a set of steps until one of the steps within the set fails.**

Loop –

- **The LOOP step repeats a sequence of child steps once for each element in an array that you specify.**
- **For example, if your pipeline contains an array of purchase-order line items, you could use a LOOP step to process each line item in the array.**
- **To specify the sequence of steps that make up the body of the loop (that is, the set of steps you want the LOOP to repeat), you indent those steps beneath the LOOP.**

- **Output array in loop - the output array property in a LOOP step specifies a variable to store the output of each iteration, resulting in a new array that combines the results of all iterations.**

Sequence –

- **You use the SEQUENCE step to build a set of steps that you want to treat as a group. The SEQUENCE step forms a collection of child steps that execute sequentially. This is useful when you want to group a set of steps as a target for a BRANCH step.**
- **This step is useful when you need to organize the flow of execution but don't require any conditions or loops for the steps inside the sequence.**
- **For example, you may execute a SEQUENCE to retrieve data, process it, and send a notification, all as part of one logical group.**

Exit –

- **The EXIT flow step exits the entire flow service, a specific parent flow step, or an iteration within a LOOP or REPEAT step and signals success or failure as part of the exit.**
- **You may use the EXIT step in situations where the flow should be stopped early, such as when an error condition is met or when further processing is unnecessary.**

Invoke –

- **The INVOKE step is used to call an external service or another flow service. It allows you to integrate with external systems, whether within the same integration server or external systems.**
- **INVOKE is used when you need to execute a service defined in webMethods or a third-party service such as an external REST or SOAP API.**

Try-Catch –

- **TRY-CATCH blocks allow for error handling in webMethods. The TRY block is used to wrap steps that might throw an exception, and the CATCH block is used to catch and handle exceptions.**
- **This is similar to try-catch-finally in programming languages like Java and is very useful for ensuring that errors in services don't cause the whole process to fail**

What are Transformers?

- **Transformers are the services you use to accomplish value transformations in the Pipeline view. You can only insert a transformer into a MAP step. You can use any service as a transformer.**

What are the different ways to invoke the service?

  - **Invoke step within webMethods Designer**
  - **Transformers in Map Step**
  - **Using remote Invoke built-in service can invoke flow service from other webMethods Environment**
  - **Using service.doInvoke() in Java services**
  - **Using webservices, consuming webservice provider operations.**
  - **Using webMethods supported protocols like http, https. Ex: http://hostname:portnumber/Invoke/namespace**
  - **Using the Publish subscribe mechanism in webMethods**
  - **In BPM using CAF application, invoking the flow service through a web service call**
  - **Using Scheduler in webMethods**
  - **Using File polling in webMethods**
  - **Using Startup & Shutdown services**

How to invoke a service from a browser?

- **Use a URL in the form: http://servername:port/invoke/folder.subFolder.subsubFolder/serviceName (the package name is not part of the URL in any way)**

What is a drop variable?

- **Drop pipeline variable is an explicit cleanup. It is a request for the pipeline to remove a variable from the available list of variables.**

JDBC –

- **JDBC (Java Database Connectivity) in webMethods is used to enable communication between webMethods Integration Server (IS) and relational databases (e.g., MySQL, Oracle, SQL Server).**
- **It allows you to query databases, insert, update, or delete data, and retrieve results through various JDBC adapters and services.**

Data Source Class –

- **MySQL: com.mysql.cj.jdbc.MysqlDataSource**
- **Oracle: oracle.jdbc.pool.OracleDataSource**

Adapter for JDBC -

- **webMethods Adapter for JDBC is an add-on to webMethods Integration Server that enables you to exchange data with relational databases through the use of a JDBC driver. The adapter provides seamless and real-time communication with the database without requiring changes to your existing application infrastructure.**
- **Using Adapter for JDBC, Integration Server clients can create and run services that execute transactions to retrieve data from and insert and update data in, relational databases.**
- **For example, you can use Adapter for JDBC to add a customer to an Oracle database based on data from another system connected to the Integration Server. Or you can use Adapter for JDBC to poll a Microsoft SQL Server database for customers that have been added to the database, and to send that data to the Integration Server to be inserted into another resource.**
- **It connects to adapter resources and initiates an operation on the resources.**

Adapter Services –

1. **Custom SQL**
2. **Select SQL**
3. **Update SQL**
4. **Insert SQL**
5. **Delete SQL**
6. **Dynamic SQL**
7. **BatchInsertSQL**
8. **BatchUpdateSQL**
9. **Stored Procedure**
10. **StoredProcedureWithSignature**
11. **Execute Service**

Diff between custom SQL & dynamic SQL?

- **In Custom SQL, you can pass inputs to your SQL query at runtime. With Dynamic SQL, you can pass your entire SQL statement, or part of your SQL statement can be passed at runtime; along with inputs to it. So basically you can build your SQL dynamically at runtime.**
- **You use Custom SQL when SQL query is fixed with input variable that are passed to the custom adapter service. You use dynamic SQL, if SQL query changes during the runtime; in this cases you prepare the sql query and pass it to dynamic adapter service in the runtime.**
- **Custom SQL and Dynamic SQL we have to write the queries explicitly. The main difference between Custom SQL and Dynamic SQL is; in Custom SQL we can give the input values at design time. In Dynamic SQL we can give the input values at run time.**
- **Custom SQL is faster than the Dynamic SQL because Custom SQL is pre-compiled (at design time) but dynamic SQL is not pre-compiled (compiled at runtime).**
- **Dynamic SQL is more versatile than the Custom SQL.**

Difference between Stored Procedure And StoredProcedureWithSignature?

- **Stored Procedure**
  - **allows calling a stored procedure with dynamic parameters, offering flexibility when parameter details are unknown.**
- **StoredProcedureWithSignature**

- requires predefined input/output parameters, providing better control and error prevention for fixed, well-defined procedures.

What is Execute Service?

- Execute Service in webMethods is used to run a service dynamically during runtime by providing its name and required inputs. It's helpful when you need to decide which service to call while the program is running.

Adapter Notifications -

- Monitor a database and notify the Integration Server when an action (not initiated by the Integration Server) occurs on a particular database table. For example, an adapter notification could notify the Integration Server when an update operation was performed in a specific database table.
- An adapter notification monitors a specified database table for changes, such as an insert, update, or delete operation, so that the appropriate Java or flow services can make use of the data, such as sending an invoice or publishing it to the Integration Server.

Adapter Notifications –

1. Insert Notification
2. Update Notification
3. Delete Notification
4. Basic Notification
5. StoredProcedureNotification
6. StoredProcedureNotificationWithSignature
7. Ordered Notification

Diff insert & basic notification?

- Insert Adapter Notification is used to retrieve the insert data from the buffer table and publish for the trigger to use it. Like others, it is also polling-based. Point is the trigger and buffer table is created automatically and dropped by IS when the notification is created and disabled (not suspended).
- In basic, you create the trigger and buffer table. So the trigger and buffer table is not created and deleted automatically when notification is disabled .

Notification Type	Notification Template	Description
Insert Notification	Insert Notification	Publishes notification of insert operations on a database table.
Update Notification	Update Notification	Publishes notification of update operations on a database table.
Delete Notification	Delete Notification	Publishes Notification of delete operations on a database table.
Basic Notification	Basic Notification	Polls a database table for data using a SQL Select operation.
Stored Procedure	Stored Procedure Notification	Publishes notification data by calling a stored procedure inside of a database.
Ordered Notification	Ordered Notification	Publishes notification data for multiple insert, update, or delete operations on multiple tables for a given database.

What is difference between local, xa and no transactions?

- Local transaction
  - means in a single transaction we can do multiple operations on a single database. To be clearer all these multiple operations use the same Adapter connection explicit commit required.
- NO Transaction
  - Auto commit all transactions.

What is start transaction commit & roll back transaction & how to use them? what do you mean by roll back transaction?

- **These statements provide control over use of transactions**
- **START TRANSACTION or BEGIN start a new transaction.**
- **COMMIT – commits the current transaction, making its changes permanent.**
- **ROLLBACK – rolls back the current transaction, cancelling its changes.**
- **ROLLBACK TRANSACTION - rolls back an explicit or implicit transaction to the beginning of the transaction, or to a save point inside the transaction. It also frees resources held by the transaction**

Type of transaction? Local ,XA,No transaction Explain?

- **NO_TXN: When you configure any JDBC Connection with NO_TXN as TXN type, the Adapter Services which use this kind of connection need not be used commit and rollback services explicitly by the developer. Which means all the transactions are auto commit or auto roll back.**
- **LOCAL_TXN: For Adapter Services which use any connection that was configured with LOCAL_TXN as TXN Type, then the developer has to use the pub.art.transaction.commit & pub.art.transaction.rollback explicitly in his code; which means the transactions are not auto committed and needs explicit transaction control.**
- **XA_TXN: Same as Local txn; but the adapter services you write in between start and end can talk to Different Databases; which means XA_TXN supports multiphase transactions. A transaction will be used when you want to say, insert content into 2 Diff DBs, one after the other; If the insert in 2nd DB fails then if you want to roll back in the 1st DB as well, then use XA txn.**

Note:

Insert Notifications, Update Notifications, Delete Notifications, and Ordered Notifications support LOCAL_TRANSACTION mode only.

Ports -

- **webMethods/HTTP**
- **webMethods/HTTPS**
- **webMethods/File Polling**
- **webMethods/FTPS**
- **webMethods/FTP**
- **webMethods/Email**
- **webMethods/WebSocket Secure**
- **webMethods/WebSocket**
- **HTTP Diagnostic**
- **HTTPS Diagnostic**
- **Enterprise Gateway Server**
- **Internal Server**

File Polling –

- **A File Polling Port monitors a specific directory for files and processes them automatically using a defined flow service.**
- **File polling in webMethods is a process where the Integration Server automatically monitors a specific folder for incoming files. When a new file is detected, it is moved to a working directory, where a specified processing service is executed. This service, which is custom-built, performs actions like parsing, transforming, or validating the file's content. Once processing is complete, the file is moved to a completion directory if successful, or to an error directory if there's an issue. The server also periodically cleans up these directories based on configurable settings, ensuring efficient file handling and integration.**

File Polling Listener Configuration -

Package -

- **Package Name**
- **Alias**
- **Description (optional) - A description of the port.**

Polling Information -

- **Monitoring Directory**
- **Working Directory (optional)**
- **Completion Directory (optional)**
- **Error Directory (optional)**
- **File Name Filter (optional)**
- **File Age (optional) (seconds)**
- **Content Type (optional)**
- **Allow Recursive Polling**
- **Enable Clustering**
- **Lock File Extension (optional)**
- **Number of files to process per Interval (optional)**

Security -

- **Run services as user**

Message Processing -

- **Enable**
- **Processing Service**
- **File Polling Interval (seconds)**
- **Log only when directory availability changes**
- **Directories are NFS mounted file system**
- **Cleanup Service (optional)**
- **Cleanup At Startup**
- **Cleanup File Age (optional) (days)**
- **Cleanup Interval (optional) (hours)**
- **Maximum Number of Invocation Threads**

Package

Defines basic metadata for the configuration.

- **Package Name: Unique identifier for the package or configuration.**
- **Alias: An alternate name or shorthand for the package.**
- **Description (optional): A human-readable explanation of the port's function or purpose.**

Polling Information

Defines how and where the system monitors for files.

- **Monitoring Directory: The main folder where the system checks for incoming files.**
- **Working Directory (optional): Temporary folder where files may be processed.**
- **Completion Directory (optional): Where successfully processed files are moved.**
- **Error Directory (optional): Where problematic or failed files are moved.**
- **File Name Filter (optional): Specifies patterns (like *.xml) to limit files being polled.**
- **File Age (optional): Minimum age in seconds before a file is eligible for processing.**
- **Content Type (optional): MIME type or format of files expected.**
- **Allow Recursive Polling: Enables polling within subdirectories.**
- **Enable Clustering: Allows this configuration to be part of a clustered environment (multi-node processing).**
- **Lock File Extension (optional): Helps avoid file conflicts by using lock files/extensions during processing.**
- **Number of files to process per Interval (optional): Limits how many files are handled in each polling cycle.**

Security

Specifies under whose authority the services run.

- **Run services as user: Indicates a specific user account (rather than system default) will execute the services for added security or permissions.**

Message Processing

Settings related to how the system handles and cleans up messages or files.

- **Enable: Activates the message processing service.**
- **Processing Service: The logic or module that processes the polled files/messages.**

- **File Polling Interval (seconds):** Frequency with which the system checks the directory for new files.
- **Log only when directory availability changes:** Limits log entries to directory status changes only (e.g., accessible vs. not accessible).
- **Directories are NFS mounted file system:** Indicates that directories are on a Network File System, possibly affecting locking or performance.
- **Cleanup Service (optional):** Optional module to remove old or processed files.
- **Cleanup At Startup:** If enabled, performs cleanup immediately on system start.
- **Cleanup File Age (optional) (days):** Deletes files older than a certain number of days.
- **Cleanup Interval (optional) (hours):** Frequency at which cleanup tasks run.
- **Maximum Number of Invocation Threads:** Limits how many parallel processing threads are used for handling messages/files.

Publish-and-Subscribe Model -

- The publish-and-subscribe model is a specific type of message-based solution in which messages are exchanged Unidentified through a message broker. Applications that produce information that needs to be shared will make this information available in specific types of recognizable documents that they publish to the message broker. Applications that require information subscribe to the document types they need.
- At run time, the message broker receives documents from publishers and then distributes the documents to subscribers. The subscribing application processes or performs work using the document and may or may not send a response to the publishing application.

Queue & Topic -

In webMethods, queues and topics are part of its messaging infrastructure, typically implemented using Universal Messaging (UM) or Broker for asynchronous communication.

Queue -

A queue is a point-to-point messaging model where a single message is sent by a producer and consumed by one consumer.

Topic -

A topic is a publish-subscribe messaging model where a single message can be consumed by multiple subscribers.

Universal Messaging –

- Universal Messaging is a Message Orientated Middleware product that guarantees message delivery across public, private, and wireless infrastructures.

Local Messaging –

- Local publishing refers to the process of publishing a document within the Integration Server. Only subscribers located on the same Integration Server can receive and process the document.

Scheduling –

- Universal Messaging provides a scheduling engine that enables tasks to be executed as server-side scripts on a server instance at specific times or when certain conditions occur within the instance.
- This enables server instances to automate important tasks, enabling them to self-manage without the need for intervention by administrators or externally scheduled tasks. The scripts consist of initial tasks, triggered tasks, and/or calendar tasks.

How will you create the Schedulers in webMethods?

- IS Admin Page - > Scheduler - > Create a scheduled task
- Run once
- Repeating tasks
- Complex Repeating

SMTP –

- **Simple Mail Transfer Protocol, an internet standard that allows mail servers to send, receive, and relay email messages.**

What is API?

- **An API for a website is code that allows two software programs to communicate with each another. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.**

REST –

- **REST (Representational State Transfer) is an architectural style that requires web applications to support HTTP methods such as GET, POST, PUT, PATCH, DELETE, and HEAD, and to use a consistent, application-independent interface. Integration Server can act as a REST server or a REST client.**

SOAP –

- **SOAP (Simple Access Object Protocol) The API details view for a SOAP API displays the details of the API such as Basic and Technical information, Operations available, REST transformation details, API mocking details, and specifications**

RestResorce –

- **In webMethods, a REST resource is an endpoint on an Integration Server that follows the principles of Representational State Transfer (REST). It allows communication between a client and the server using standard HTTP methods such as GET, POST, PUT, and DELETE**

Rest API Descriptor –

- **A REST API descriptor provides a way of describing the operations provided by one or more REST resources together with information about how to access those operations, the MIME types the resources consume and produce, and the expected input and output for the operations. Fundamentally, a REST API descriptor is composed of REST resources and information about how to access those resources**

How do you create a REST API in webMethods?

- **Create a REST Resource, define HTTP methods (GET, POST, PUT, DELETE), and implement services for each method. Use the API Gateway for management.**

1. **New -> REST Resource**
2. **Select the resource type**
3. **And select the check box for generate JSON API URL templates and click Finish**

How is a SOAP API created in webMethods?

- **Generate a WSDL in Designer, implement the corresponding Flow services, and expose them as SOAP services via an endpoint.**

1. **New-> webservice descriptor**
2. **Select the description -> Provider / consumer**
3. **If it is provider, select the list of service which we want to choose and click Finish. (each service will be exposed**
4. **as a operation) and download the WSDL and start using it from external app.**
5. **If it is consumer, then browse the WSDL details to consume the webservice and click finish.**

What are the key differences between REST and SOAP APIs?

- **REST is stateless and uses lightweight formats like JSON, while SOAP is stateful and uses XML with extensive WS-* standards.**

SOAP	REST
1)An XML-based message protocol.	An Architecture Style Protocol.
2) Use WSDL for communication between consumer and provider.	Uses XML or JSON to send and receive data.

3) Invoking service by calling RPC methods.	Simply call the service via the URL path.
4) Does not return human-readable format.	The result is readable, which is just plain XML or JSON.
5) Transfer Over HTTP. Also use other protocols such as SMTP, FTP, etc	A transfer is over HTTP only.
6)Java Script can be called SOAP, But it is difficult to implement.	Easy to call from JavaScript
7) Performance is not great.	Performance is much better, Less CPU use, Leaner code, Lightweight, etc.

Swagger –

- In webMethods, Swagger is used to design, document, and test REST APIs. It supports the OpenAPI Specification (OAS), enabling you to import/export API definitions in JSON or YAML. Swagger UI provides interactive documentation for testing APIs directly from a browser.

What is WSDL ? What does the WSDL file contains ?

- WSDL (Web Services Description Language) in webMethods is an XML-based standard used to describe web services. It serves as a contract between the service provider and the consumer, defining the service's functionality, operations, messages, and data types.
- WSDL – Webservices Description Language - is an XML specification for SOAP-based webservice.
- It contains the request structure, response structure, and URL endpoint details required for SOAP webservice implementation.

ACL –

- Access Control List (ACL) in webMethods is a security mechanism used to manage and control access to resources such as services, folders, and packages within the webMethods Integration Server. ACLs allow you to define which users or groups can perform specific actions, ensuring secure and restricted access to sensitive or critical resources.

Assest Properties –

- Asset Properties are configurations that define the characteristics or attributes of an asset, such as services, adapters, and integration components. These properties are used to manage and control the behavior of an asset.

Transition –

- A transition line can be modified to change its route, colour, line format, and the font characteristics of the associated text.

Transition Type –

- In webMethods, transition types define the logical behaviour of a transition that routes information from one step to another in a process mode.

Data Storage –

- For a publishable document type, you can set the storage type to determine how Integration Server and the messaging provider store instances of this document. The storage type also determines how quickly the document moves through the webMethods system. You can select one of the following storage types:

- **Volatile storage**
  - specifies that instances of the publishable document type are stored in memory. Volatile documents move through the webMethods system more quickly than guaranteed documents because resources do not return acknowledgements for volatile documents.
  -
- **Guaranteed storage**
  - specifies that instances of the publishable document type are stored on disk. Resources return acknowledgements after storing or processing guaranteed documents. Because guaranteed documents are saved to disk and acknowledged,

guaranteed documents move through the webMethods system more slowly than volatile documents.

Document Type Storage Types –

- The document storage type setting determines how instances of that document are persisted. Documents published to Broker can be stored as *volatile* documents or *guaranteed* documents.

- **Volatile documents** -
  - are stored in local memory only. These documents are lost if the Broker host experiences a service interruption or the Broker Server is restarted. To reduce memory usage, volatile documents that have expired can be proactively deleted at regular intervals, based on the size of the queue, from the client queues and forward queues before the client tries to retrieve them.

- **Guaranteed documents**
  - are persisted to disk so that they can be recovered in the event of a power failure or a server restart. This is the default storage type for document types. Guaranteed storage provides lower performance than volatile storage, but very little risk of losing events if a hardware, software, or network failure occurs. This type of storage is the safest and is suited for documents that you do not want to lose.

- **Implicit map**
  - - Designer implicitly links variables whose names are the same and whose datatypes are compatible. Designer connects implicitly linked variables with a Gray link.

- **Explicit map**
  - In cases where the services in a flow do not use the same names for a piece of information, use the Pipeline tab to explicitly link the variables to each other. It will have solid black line.

How to handle exceptions in webMethods?

- To handle exceptions in webMethods, we use three sequence steps – two child sequence steps nested under a parent sequence step as below

  First sequence (exit on success)

  Second sequence (exit on failure)

  Third sequence (exit on done)

- In the new version, we can use try/catch block flow step

What is a specification?

- A specification is a IS element that defines a set of service inputs and outputs. If you have multiple services with the same input and output requirements, you can point each service to a single specification rather than manually specify individual input and output fields in each service.

What is the difference between an adapter service and an adapter notification?

- Adapter services are for database operations (e.g., Insert, Update), while adapter notifications capture database events (e.g., record inserts) to trigger services.

What are Keystore and Truststore?

- **Keystore:**
  - The Integration Server stores its private keys and SSL certificates in keystore files.

- **Truststore:**

- Integration Server uses a truststore to store its trusted root certificates, which are the public keys for the signing CAs.It simply functions as a database containing all the public keys for CAs within a specified trusted directory.

Explain Triggers in detail.

- In webMethods, a trigger is like a listener that waits for specific messages (called documents) to arrive. When the message comes, the trigger picks it up and sends it to a service to process. This helps different systems communicate without waiting for each other.
- In webMethods, a trigger is a mechanism that listens for publishable documents from a messaging system (e.g., Universal Messaging) and routes them to a specified service for processing, enabling asynchronous communication in the publish-subscribe model.
- Triggers establish subscriptions to publishable document types and specify how to process instances of those publishable document types. When you build a trigger, you create one or more conditions. A condition associates one or more publishable document types with a single service. The publishable document type acts as the subscription piece of the trigger. The service is the processing piece. When the trigger receives documents to which it subscribes, the Integration Server processes the document by invoking the service specified in the condition. Triggers can contain multiple conditions.

  The trigger contains at least one condition.

  Each condition in the trigger specifies a unique name.

  Each condition in the trigger specifies a service.

  Each condition in the trigger specifies one or more publishable document types.

JMS triggers –

- A JMS trigger is a trigger that receives messages from a destination (queue or topic) on a JMS provider and then processes those messages.

VETO pattern in webMethods Integration –

- The VETO pattern in webMethods can be explained simply as:
- V - Validation: Check if the data is correct and complete.
- E - Enrichment: Add any missing information to the data.
- T - Transformation: Change the data into the format the target system needs.
- O - Orchestration: Control the flow of processes and decide where the data should go next.
- It's like cleaning, completing, converting, and directing data!

What is DSP?

- In webMethods, DSP (Dynamic Server Pages) is a technology used to create dynamic web pages that integrate with backend services. It allows incorporating webMethods logic into HTML, enabling the creation of custom user interfaces and interactive web applications.

Integration pattern in webMethods?

- In webMethods, integration patterns refer to the architectural approaches used to connect disparate systems, applications, and services efficiently.
- Integration patterns include Point-to-Point for direct connections, Hub-and-Spoke for centralized communication, Publish-Subscribe for event-driven messaging, API-Led for reusable APIs, and File-Based or Batch integrations for bulk data processing.

What is Package?

A package is a container that is used to bundle services and related elements, such as specifications, IS document types, IS schemas, and triggers. When you create a folder, service, IS document type, or any element, you save it in a package.

What is Flow Service?

A package is a container that is used to bundle services and related elements, such as specifications, IS document types, IS

schemas, and triggers. When you create a folder, service, IS document type, or any element, you save it in a package.

What is Flow Step?

A flow service contains flow steps. A flow step is a basic unit of work (expressed in the webMethods flow language) that webMethods Integration Server interprets and executes at run time. The webMethods flow language provides flow steps that invoke services and flow steps that let you edit data in the pipeline.

What is Pipeline?

The pipeline is the general term used to refer to the data structure in which input and output values are maintained for a flow service. It allows services in the flow to share data.

What is Adapter Service?

Adapter services allow you to connect to the adapter's resource and initiate an operation on the resource from Integration Server. You call adapter services from flow or Java services to interact with database tables. Integration Server then uses adapter connections that you defined earlier to execute the adapter services. Adapter services are based on templates provided with Adapter for JDBC. Each template represents a specific technique for doing work on a resource, such as using the SelectSQL template to retrieve specified information from a database.

What is Adapter Notification?

An adapter notification contains information about an event that occurs on an adapter resource and then sends the notification data to Integration Server in the form of a published document.

What is Universal Messaging?

WebMethods Universal Messaging is a message-orientated middleware product that guarantees message delivery across public, private, and wireless infrastructures. Universal Messaging has been built from the ground up to overcome the challenges of delivering data across different networks. It provides its guaranteed messaging functionality without the use of a web server or modifications to firewall policy.

What is Flat File?

Flat files present complex hierarchical structural data in a record–based storage format. Unlike XML, flat files do not embed structural data (metadata) within the data. The data in the flat file has been “flattened” by removing the hierarchical relationship between records, leaving the records intact as a single logical record of application data.

What is Flat File Schema?

A flat file schema is the blueprint that contains the instructions for parsing or creating a flat file and is created as a namespace element in the Integration Server. This blueprint details the structure of the document, including delimiters, records, and repeated record structures. A flat file schema also acts as the model against which you can validate an inbound flat file.

What is Flat File Dictionary?

A flat file schema can contain either record definitions or references to record definitions that are stored elsewhere in the namespace in a flat file dictionary. A flat file dictionary is simply a repository for elements that you reference from flat file schemas. This allows you to create record definitions in a dictionary that can be used across multiple flat file schemas. Reusing record definitions reduces the amount of memory consumed by a flat file schema.

What is Trigger?

Trigger is a powerful tool that automatically launches a workflow when a defined event happens. This enables you to automate a complex business process without having to manually run the workflow every time.