WEB SERVICES WITH JAX-WS AND JAX-RS

- R. KISHORE PRABHU

1531019

CONTENTS

INTRODUCTION TO WEB SERVICES

Web service is a means by which computers talk to each other over the web using HTTP and other universally supported protocols.

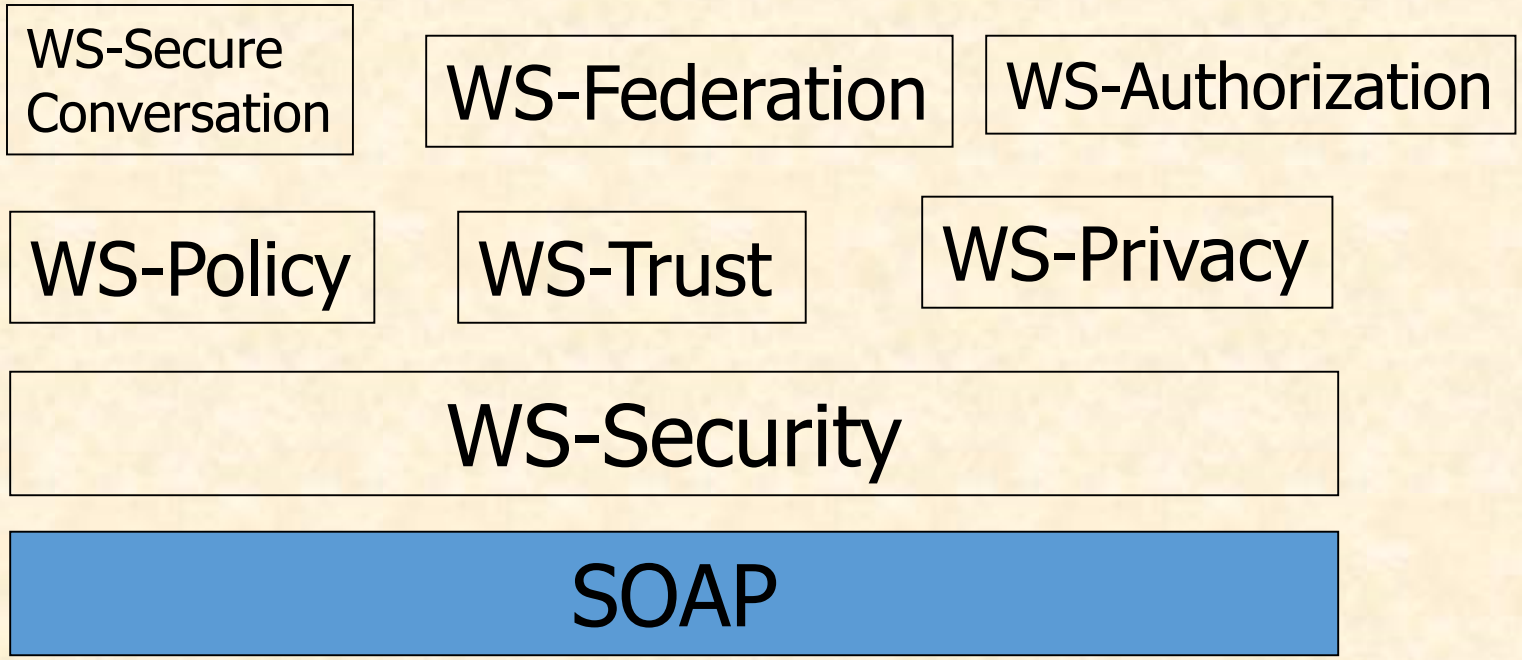
A Web service is an application that:

- Runs on a Web server
- Exposes Web methods to interested callers
- Listens for HTTP requests representing commands to invoke Web methods
- Executes Web methods and returns the results

SOAP Web Services

- SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment.
- SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols.
- The framework has been designed to be independent of any particular programming model and other implementation-specific semantics.
- **SOAP is a way for a program running in one operating system to communicate with a program running in either the same or a different operating system, using HTTP (or any other transport protocol) and XML.**

SOAP is the Foundation



JAX-WS Client

- Stand-alone program that accesses the server methods.
- This call is made through a port, a local object that acts as a proxy for the remote service.
- This port is created by wsimport, which generates JAX-WS portable artifacts based on a WSDL file.

RESTful WEB SERVICES

- Resources are identified by uniform resource identifiers (URIs)
- Resources are manipulated through their representations
- Messages are self-descriptive and stateless
- Multiple representations are accepted or sent
- Hypertext is the engine of application state

Bottom-up and Top-down Web Services

TOP-DOWN:

- **Top-down Web services development** involves creating a WSDL and then creating the Web service using the WSDL file.
  - In top-down approach, first you design the implementation of the Web service by creating a WSDL file. You can then create the Web service skeleton Java classes from the wsdl, and add the required code.
- JAX-WS tool **wsimport** can be used for creating a top down web service.
  - The wsimport can create stubs from a wsdl.
  - *See notes on JAX-WS tools wsimport and wsgen.*
- Top down approach is also called as **contract first or WSDL first** approach.
- Most common in production scenarios.
- **Example scenarios:**
  - Users will be responsible for developing their own clients using different frameworks, based on a published WSDL.

Bottom-up and Top-down Web Services

BOTTOM-UP:

- When creating a Web service using a **bottom-up approach**, first you write the java classes for the web service and then create the WSDL file and publish the web service.
  - Although bottom-up Web service development may be faster and easier, the top-down approach is the recommended way of creating a Web service.
- JAX-WS tool **wsgen** can be used for creating a bottoms up web service.
  - The wsgen tool reads an existing web service implementation class (SIB) and generates the required JAX–WS portable artifacts for web service development and deployment.
  - *See notes on JAX-WS tools wsimport and wsgen.*
- Bottum up approach is also called as **Service first or code first** approach.
- Ideal for learning web services.
- Example scenarios:
  - Situations which need quick and faster completion may prefer Java approach

WEB SERVICES ERROR HANDLING

HTTP defines over 40 standard status codes that can be used to convey the results of a client's request. The status codes are divided into the five categories presented here:

- **1xx: Informational** - Communicates transfer protocol-level information
- **2xx: Success** -Indicates that the client's request was accepted successfully.
- **3xx: Redirection** - Indicates that the client must take some additional action in order to complete their request.
- **4xx: Client Error** - This category of error status codes points the finger at clients.
- **5xx: Server Error** - The server takes responsibility for these error status codes.

SECURITY CONCEPTS

- **Authentication**—Verifying that the user is who she claims to be. A user's identity is verified based on the credentials presented by that user, such as:
  - Something one has, for example, credentials issued by a trusted authority such as a passport (real world) or a smart card (IT world).
  - Something one knows, for example, a shared secret such as a password.
  - Something one is, for example, biometric information.
- Using a combination of several types of credentials is referred to as "strong" authentication, for example using an ATM card (something one has) with a PIN or password (something one knows).
- **Authorization (or Access Control)**—Granting access to specific resources based on an authenticated user's entitlements. Entitlements are defined by one or several attributes. An attribute is the property or characteristic of a user, for example, if "Marc" is the user, "conference speaker" is the attribute.
- **Confidentiality, privacy**—Keeping information secret. Accesses a message, for example a Web service request or an email, as well as the identity of the sending and receiving parties in a confidential manner. Confidentiality and privacy can be achieved by encrypting the content of a message and obfuscating the sending and receiving parties' identities.
- **Integrity, non repudiation**—Making sure that a message remains unaltered during transit by having the sender digitally sign the message. A digital signature is used to validate the signature and provides non-repudiation. The timestamp in the signature prevents anyone from replaying this message after the expiration.

Thank You..