Proyecto TIC: Desarrollo e implantación

Web Services
Java EE.

Xavi Piñol
xavi.pinyol@gmail.com

Index

1. What are Web Services?
2. Types of Web Services.

Index

WS. What are Web Services?

Web services are client and server applications that communicate over the World Wide Web's (WWW) HyperText Transfer Protocol (HTTP). Web services provide a standard means of interoperating between software applications running on a variety of platforms and frameworks. Web services are characterized by their interoperability and extensibility. Web services can be combined in a loosely coupled way to achieve complex operations. Programs providing simple services can interact with each other to deliver sophisticated added-value services.

WS. What are Web Services?

Many organizations use multiple software systems for management. Different software systems often need to exchange data with each other, and a Web service is a method of communication that allows two software systems to exchange this data over the internet. The software system that requests data is called a **service requester**, whereas the software system that would process the request and provide the data is called a **service provider**.

WS. What are Web Services?

Different software may use different programming languages, and hence there is a need for a method of data exchange that doesn't depend upon a particular programming language. Most types of software can, however, interpret XML tags. Thus, Web services can use XML files for data exchange.

Web services can be used on Internet and on intranets.

Index

WS. Types of Web Services

On the conceptual level, a service is a software component provided through a network-accessible endpoint. The service consumer and provider use messages to exchange invocation request and response information in the form of self-containing documents that make very few assumptions about the technological capabilities of the receiver.

On a technical level, web services can be implemented in various ways. The two types of web services discussed in this section can be distinguished as “SOAP” web services and “RESTful” web services.

Index

WS. SOAP Web Services.

In Java EE 6, JAX-WS provides the functionality for “Soap” web services. This type of web services use XML messages that follow the Simple Object Access Protocol (SOAP) standard, an XML language defining a message architecture and message formats. Such systems often contain a machine-readable description of the operations offered by the service, written in the Web Services Description Language (WSDL), an XML language for defining interfaces syntactically.

WS. SOAP Web Services.

The SOAP message format and the WSDL interface definition language have gained widespread adoption. Many development tools, such as Jboss Developer Studio, can reduce the complexity of developing web service applications.

WS. SOAP Web Services.

A formal contract must be established to describe the interface that the web service offers. WSDL can be used to describe the details of the contract, which may include messages, operations, bindings, and the location of the web service.

Index

WS. RESTFul Web Services.

In Java EE 6, JAX-RS provides the functionality for Representational State Transfer (RESTful) web services. REST is well suited for basic, ad hoc integration scenarios. RESTful web services, often better integrated with HTTP than SOAP-based services are, do not require XML messages or WSDL service–API definitions.

WS. RESTFul Web Services.

A RESTful design may be appropriate when the following conditions are met:

- The web services are completely stateless. A good test is to consider whether the interaction can survive a restart of the server.

WS. RESTFul Web Services.

- A caching infrastructure can be leveraged for performance. If the data that the web service returns is not dynamically generated and can be cached, the caching infrastructure that web servers and other intermediaries inherently provide can be leveraged to improve performance. However, the developer must take care because such caches are limited to the HTTP GET method for most servers.

WS. RESTFul Web Services.

- The service producer and service consumer have a mutual understanding of the context and content being passed along. Because there is no formal way to describe the web services interface, both parties must agree out of band on the schemas that describe the data being exchanged and on ways to process it meaningfully.

WS. RESTFul Web Services.

- Bandwidth is particularly important and needs to be limited. REST is particularly useful for limited-profile devices, such as mobile phones, for which the overhead of headers and additional layers of SOAP elements on the XML payload must be restricted.

WS. RESTFull Web Services.

- Web service delivery or aggregation into existing web sites can be enabled easily with a RESTful style. Developers can use such technologies as JAX-RS and Asynchronous JavaScript with XML (AJAX) to consume the services in their web applications. Rather than starting from scratch, services can be exposed with XML and consumed by HTML pages without significantly refactoring the existing web site architecture. Existing developers will be more productive because they are adding to something they are already familiar with rather than having to start from scratch with new technology.