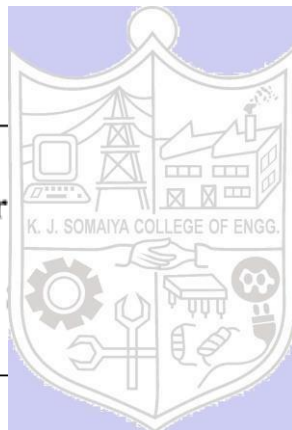


Exper

Title:



on Web Services

Batch: A2

Roll No.: 16010421063

Name: Arya Nair

Experiment No.:

9 Aim: Study Experiment on Web Services**Theory:****Web Services:**

Web services are software mechanisms that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging and this structure are based on a collection of standards and protocols that allow us to make handling requests to remote systems by delivering a standard, nonproprietary language and using conventional transport protocols such as HTTP and SMTP. The efficient e-business perception calls for a smooth integration of business processes, applications, and Web services over the Internet. Web service technology enables e-business and e-commerce to become a reality. It has become a competitive tool for companies by reducing cost through fast, efficient, and reliable services to clients, dealers, and partners over the Internet. It permits more efficient business processes via the Web and improves business chances for companies, Web services are planned to be accessed by other applications and differ in complication from primary activities, such as examine a banking account balance online, to complicated processes running CRM (customer relationship management) or enterprise resource planning (ERP) systems because these are based on open standards such as HTTP and XML-based protocols including SOAP and WSDL. Web services are powered by XML and three other core technologies: WSDL, SOAP, and UDDI. In a Web service model, a service supplier proposals Web services which deliver tasks or business operations which can be arranged over the Internet, in the hope that they will be invoked by partners or customers; a Web service requester defines requirements to trace service provider. Publishing, binding, and discovering Web services are three key tasks in the model. Discovery is the process of finding Web services provider locations which satisfy specific requirements. Web services are useless if they cannot be discovered. So, discovery is the most important task in the Web service model. The Web service model in Figure shows the interaction between a service requester, service providers, and a service discovery system.

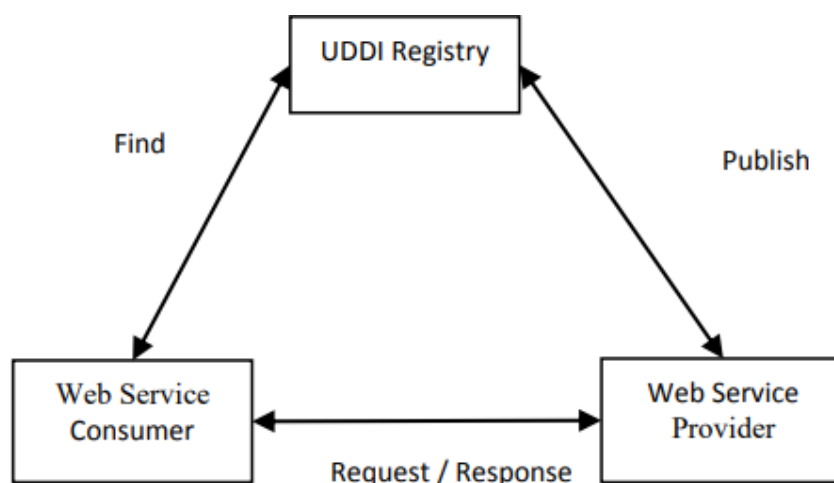


Figure: Web Services Model

1. The service provider's proposal Web services which deliver functions or business operations. They are formed by companies or societies. In order to be invoked, the Web services must be defined. This will facilitate discovery and arrangement. WSDL or service profile of semantic Web service is used to carry out this task.
2. The Web service requester defines requirements in order to locate service providers. Service requesters usually contain a description of the Web service, though it is not a Web service which can run on the Internet. The requirements are typically defined by WSDL, service template or service profile.
3. The Web service discovery or service registry is a broker that provides registry and examine tasks. The service providers advertise their service info in the discovery system. This info will be kept in the registry and will be searched once there is a demand from service requester. UDDI is used as a registry typical for Web service.

The above three mechanisms interact with each other via publishing, discovery, and binding operations. These operations are elaborated upon as follows:

1. **Publish:** the Web service providers publish their service information through the discovery system for requesters to discover. Through the publishing operation, the Web service provider stores the service description in the discovery system.
2. **Discovery:** the Web service requesters repossess service providers from the service archive. Based on service explanations, which describes the requests of the Web service clients, the discovery system will output a list of Web service suppliers which satisfy the requirements.
3. **Bind:** After discovering, the discovery system provides some Web service providers. The Web service requester invokes these Web service providers. The binding occurs at runtime. The Web service requesters and Web service providers will communicate via SOAP protocol which is an XML based protocol for Web service exchange information.

Activity:

1. Case study of Google web services.

Google is one of the most widely used search engines in the world, with over 90% of internet users utilizing the platform. However, Google also offers a range of other web services that are popular among both individuals and businesses. In this case study, we will take a closer look at some of Google's most popular web services, including Google Search, Google Maps, Google Drive, and Gmail.

1. Google Search: Google Search is the flagship web service of Google, allowing users to search for information on the web. Google uses a complex algorithm to sort through the billions of pages on the web to provide users with the most relevant results. Google's search algorithm takes into account a wide range of factors, including the content of the webpage, the number and quality of links pointing to the webpage, and the user's search history and location. Google Search uses HTML to display search results to users. When a user enters a search query, Google's search engine uses XML to parse and extract information from the webpages on the internet. Google then uses its complex algorithm to rank and sort the search results. However, Google Search does not use WSDL, SOAP, or UDDI.

Publishing: Google Search publishes its search engine results pages (SERPs) through HTML pages, which can be accessed by any web browser. Google also provides APIs for developers to integrate search functionality into their applications. However, these APIs are limited in functionality and are not publicly available for general use.

Discovery: Google Search uses a proprietary algorithm to rank and sort its search results, so there

is no formal discovery mechanism for its web service.

Binding: Google Search does not use any formal protocol for binding. Instead, search results are displayed in HTML pages that can be accessed directly by users or by developers through web scraping.

2. Google Maps: Google Maps is a web mapping service that provides users with detailed maps, satellite imagery, and street view imagery of locations all around the world. Google Maps also offers features such as turn-by-turn navigation, real-time traffic updates, and public transit information. Google Maps uses a combination of satellite imagery, aerial photography, and street-level photography to create its maps.

Google Maps uses HTML, XML, and JavaScript to display maps and related information to users. The service uses a combination of XML and HTTP requests to retrieve data from its servers and to display the data on the user's device. Google Maps does not use WSDL or SOAP, but it uses UDDI to provide information about its services.

Publishing: Google Maps provides APIs for developers to integrate maps, location data, and other related information into their applications. These APIs use a RESTful architecture that enables developers to easily communicate with Google's servers. The APIs are publicly available and are extensively documented to assist developers in their use.

Discovery: Google Maps publishes its web services in the UDDI registry, making it easy for developers to find and use them. The UDDI registry contains detailed information about the APIs, including their functionality and usage guidelines.

Binding: Google Maps APIs use a specific URL structure and endpoint that developers can use to bind their applications to the service. Developers can also use OAuth 2.0 to authenticate their applications when accessing the API. By using these standards, Google Maps makes it easy for developers to integrate its web services into their applications.

3. Google Drive: Google Drive is a cloud storage and collaboration platform that allows users to store and share files online. Google Drive offers users 15GB of free storage space, with additional storage available for purchase. Users can access their files from any device with an internet connection and can collaborate on files in real-time with other users.

Google Drive uses a RESTful API that is based on HTTP and JSON to communicate with its servers. The service also uses HTML and JavaScript to display the web interface to users. Google Drive does not use WSDL, SOAP, or UDDI.

Publishing: Google Drive provides APIs for developers to access and manage files stored on Google Drive. These APIs use a RESTful architecture and are publicly available for developers to use. Google also provides SDKs for popular programming languages, making it easier for developers to integrate their applications with Google Drive.

Discovery: Google Drive APIs are published in the Google Developers Console, which provides a portal for developers to register and manage their applications. The Developers Console also provides extensive documentation on the APIs, as well as usage guidelines and support resources.

Binding: Google Drive APIs use a specific URL structure and endpoint that developers can use to bind their applications to the service. Developers can also use OAuth 2.0 to authenticate their applications when accessing the API. By using these standards, Google Drive makes it easy for developers to integrate its web services into their applications.

4. Gmail: Gmail is a free email service offered by Google that allows users to send and receive email messages. Gmail offers a wide range of features, including spam protection, customizable themes, and integration with other Google services such as Google Drive and Google Calendar. Gmail also offers a robust search feature that allows users to quickly find specific emails.

Gmail uses HTML and JavaScript to display the user interface to users. The service uses a

combination of XML and HTTP requests to communicate with its servers.

Publishing: Gmail provides APIs for developers to access and manage email data in their applications. These APIs use a RESTful architecture and are publicly available for developers to use. Google also provides SDKs for popular programming languages, making it easier for developers to integrate their applications with Gmail.

Discovery: Gmail APIs are published in the Google Developers Console, which provides a portal for developers to register and manage their applications. The Developers Console also provides extensive documentation on the APIs, as well as usage guidelines and support resources.

Binding: Gmail APIs use a specific URL structure and endpoint that developers can use to bind their applications to the service. Developers can also use OAuth 2.0 to authenticate their applications when accessing the API. By using these standards, Gmail makes it easy for developers to integrate its web services into their applications.

Conclusion: Hence a case study on google services was studied.

References:

1. <https://www.geeksforgeeks.org/what-are-web-services/>
2. https://www.tutorialspoint.com/webservices/what_are_web_services.htm

