Flask Question

1) How is memory managed in Flask?

Flask maintains a collection of private heap spaces which holds all the objects and data structures together. As a developer we cannot handle it. It is the task of the interpreter to manage it. It also has an in built garbage collector which recycles all the unused memory so that it is available for the heap space to allocate new data and objects.

2) Difference between post and get request?

Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to the server.

3) Difference between Rest and Soap API?

SOAP (Simple Object Access Protocol) is a protocol which was designed before REST and came into the picture. The main idea behind designing SOAP was to ensure that programs built on different platforms and programming languages could exchange data in an easy manner.

REST - This was designed specifically for working with components such as media components, files, or even objects on a particular hardware device. Any web service that is defined on the principles of REST can be called a RestFul web service. A Restful service would use the normal HTTP verbs of GET, POST, PUT and DELETE for working with the required components.

KEY DIFFERENCE

SOAP stands for Simple Object Access Protocol whereas REST stands for Representational State Transfer.

SOAP is a protocol whereas REST is an architectural pattern.

SOAP uses service interfaces to expose its functionality to client applications while REST uses Uniform Service locators to access to the components on the hardware device.

SOAP needs more bandwidth for its usage whereas REST doesn’t need much bandwidth.

SOAP only works with XML formats whereas REST work with plain text, XML, HTML and JSON.

SOAP cannot make use of REST whereas REST can make use of SOAP.

|  |  |
| --- | --- |
| SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern. | REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern. |

SOAP requires more bandwidth for its usage. Since SOAP Messages contain a lot of information inside of it, the amount of data transfer using SOAP is generally a lot.

<?xml version="1.0"?>

<SOAP-ENV:Envelope

xmlns:SOAP-ENV

="http://www.w3.org/2001/12/soap-envelope"

SOAP-ENV:encodingStyle

=" http://www.w3.org/2001/12/soap-encoding">

<soap:Body>

<Demo.guru99WebService

xmlns="http://tempuri.org/">

<EmployeeID>int</EmployeeID>

</Demo.guru99WebService>

</soap:Body>

</SOAP-ENV:Envelope>

REST does not need much bandwidth when requests are sent to the server. REST messages mostly just consist of JSON messages. Below is an example of a JSON message passed to a web server. You can see that the size of the message is comparatively smaller to SOAP.

{"city":"Mumbai","state":"Maharastra"}

|  |  |
| --- | --- |
| SOAP uses service interfaces to expose its functionality to client applications. In SOAP, the WSDL file provides the client with the necessary information which can be used to understand what services the web service can offer. | REST use Uniform Service locators to access to the components on the hardware device. For example, if there is an object which represents the data of an employee hosted on a URL as http://demo.guru99 , the below are some of URI that can exist to access them  http://demo.guru99.com/Employee  http://demo.guru99.com/Employee/1 |

4) Is REST a Stateless or Stateful?

In **Stateless**, server is not needed to keep the server information or session details to itself. In **stateful**, a server is required to maintain the current state and session information. In **stateless**, server and client are loosely coupled and can act independently. In **stateful**, server and client are tightly bound.

**REST is stateless**, the client context is not stored on the server between requests, giving **REST** services the ability to be retried independently of one another.

5) How is Session maintained is REST is stateless and nothing is getting stored on the server?

Session is maintained on the client side (web browser) and not the server side. In stateless no information is stored on the server side.

6) What is Load balancing?

Load balancing refers to the process of distributing a set of tasks over a set of resources, with the aim of making their overall processing more efficient. This is mainly done on the server side.

7) What is Containerization?

Containerization involves bundling an application together with all of its related configuration files, libraries and dependencies required for it to run in an efficient and bug-free way across different computing environments. The most popular containerization ecosystems are Docker and Kubernetes.