1. O que é um Web Service e qual é o seu propósito principal na arquitetura de software?

O Web Service é um projeto de software que garante a interoperabilidade entre dois aplicativos em uma rede. Ele tem padrões definidos que facilitam o bom andamento das conexões.

1. Como os Web Services diferem das APIs tradicionais em termos de comunicação entre sistemas?

O Web Service são uma API que utiliza padrões claros e bem definidos. O SOAP, em formato XML e o Rest com os métodos HTTP (Get/Post/Put/Delete). As outras API’s podem utilizar outros padrões diversos.

1. Quais são os principais tipos de Web Services e como eles se diferenciam uns dos outros?

Os principais tipos são o SOAP(Em XML) e os métodos HTTP(Get/Post/Put/Delete).

1. Explique o que é uma API RESTful e por que ela é popular na criação de aplicativos web modernos.

O API RESTful é uma API baseada nos princípios Rest. Utilizando os métodos HTTP, que são realizados em URL. Ele é utilizado por ter escabilidade, simplicidade e fácil uso.

1. Quais são os principais métodos HTTP utilizados em APIs RESTful e qual é a finalidade de cada um?

Os métodos são: Get (buscar dados de um recurso), Post (Criar um novo recurso), Put (Atualiza um recurso que existe), Delete (Apaga o recurso).

1. O que é o formato de dados JSON e por que ele é amplamente utilizado em APIs?

Json é um formato de fácil transferência de dados. Ele é compatível com muitas linguagens e por isso ele é o mais utilizado. Além de ser facilmente entendido por humanos e máquinas.

1. Explique o que é o protocolo SOAP e em que contextos ele é preferível em relação às APIs RESTful.

O protocolo SOAP possui formado XML para troca de informações estruturadas e ele é preferível quando o sistema necessita de maior segurança na proteção dos dados. Como por exemplo, sistema bancário.

1. Quais são os principais padrões de segurança utilizados em Web Services e APIs para proteger a integridade e confidencialidade dos dados?

Padrão HTTPS, autenticação por Tokens e padrão OAUTH.

1. Como funciona o processo de autenticação e autorização em APIs e por que são cruciais para a segurança da informação?

A autenticação tem a ver com a identificação do usuário ou aplicativo. E a autorização é o que permite ou não a manipulação e acesso a determinados dados. É importante para que o sistema não seja invadido, causando estragos. Além de permitir apenas usuários autorizados para poder manipular dados.

1. De que forma os Web Services e APIs facilitam a integração entre diferentes sistemas e aplicativos, promovendo a interoperabilidade?

Através de padrões bem definidos, ele é capaz de se comunicar harmoniosamente com diversos softwares. É como se fosse um “tradutor”, facilitando a interoperabilidade na troca de dados de forma coesa.

1. Explique a diferença entre autenticação e autorização em contextos de Web Services e por que ambas são importantes para a segurança.

A autenticação tem a ver com a identificação do usuário, software e etc. Basicamente para saber se quem está tentando entrar no sistema é de fato ela mesma. A autorização define o que o usuário/software e etc pode ter acesso e/ou manipular dados. É fundamental ter este controle para evitar acesso de usuários não autorizados e também para que seja controlado quem pode manipular/acessar determinados dados.

1. Quais são as práticas recomendadas para proteger um Web Service contra ataques?

Utilizar métodos de autenticação forte. Permitir o mínimo de autorização a usuários, controlando as entradas para evitar golpes e definindo políticas de autorizações claras para definir o que cada usuário terá acesso e o que ele pode fazer.

1. O que faz os linhas de comando abaixo?

import javax.jws.WebMethod;

import javax.jws.WebService;

@WebService

public class HelloWorldWebService {

@WebMethod

public String sayHello(String name) {

return "Hello, " + name + "!";

}

}

Cria um Web Service SOAP.

1. De acordo com o código abaixo, qual o caminho (url) executável para chamar o serviço criado?

public class CalculatorWebServicePublisher {

public static void main(String[] args) {

String url = "http://localhost:8080/calculator"; // Endereço de acesso ao Web Service

CalculatorWebService calculator = new CalculatorWebServiceImpl();

Endpoint.publish(url, calculator);

System.out.println("Calculator Web Service está rodando: " + url);

http://localhost:8080/calculator