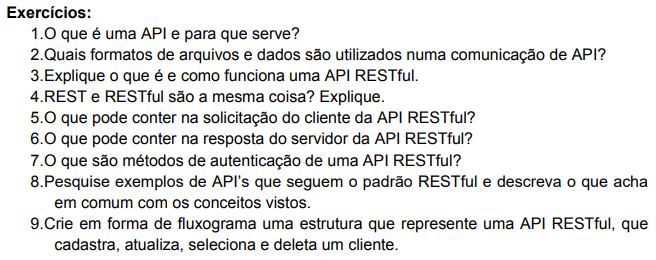
**Mod09 – servidores Atividade 07 – API RESTful**

**VERSÃO EXPANDIDA**



Q1) As interfaces de programação de aplicativos (APIs) são conjuntos de ferramentas, definições e protocolos para a criação de aplicações de software. APIs conectam soluções e serviços, sem a necessidade de saber como esses elementos foram implementados. As APIs tem a função de conectar sistemas, softwares e aplicativos. As APIs permitem que o usuário final utilize uma ferramenta sem que seja necessário acessá-la diretamente. Essa ferramenta pode ser um aplicativo, um software ou até uma simples planilha. O intuito de uma API é trocar dados entre sistemas diferentes, na maior parte das vezes essas trocas de dados tem como objetivo automatizar processos manuais e/ou permitir a criação de novas funcionalidades.

As APIs funcionam como se fossem contratos, com documentações que representam um acordo entre as partes interessadas. Se uma dessas partes enviar uma solicitação remota estruturada de uma forma específica, isso determinará como a aplicação da outra parte responderá.

Por exemplo, quando um software de CRM (Customer Relationship Management ou gerenciamento de relacionamento com o cliente, em português) tem um botão ao lado do número de telefone que permite realizar uma chamada via integração com uma provedora de telefonia VOIP e deixar gravada a chamada. Nesse caso acontece troca de dados para criar uma nova funcionalidade onde todo o serviço é realizado por uma outra empresa (provedora de telefonia).

Outro exemplo de serventia de uma API é o de um funcionário que precisa emitir notas e boletos para finalizar o pedido de um cliente. Nesse caso, a API pode conectar o sistema de gestão da empresa ao sistema de geração de boletos do banco – e ao de emissão de notas da prefeitura. Assim, o colaborador precisa apenas inserir os dados uma vez e finalizar o processo com poucos cliques.

As vantagens da automação no atendimento ao cliente e da automatização de processos trazidas pelas APIs são refletidas nas mais diversas áreas.

A seguir, apresenta-se alguns exemplos de API:

API WhatsApp (WhatsApp Business API) permite que a empresa se comunique em larga escala na plataforma que é utilizada por bilhões de pessoas no mundo inteiro.

API de voz permite a comunicação em larga escala com clientes e consumidores, através do envio de torpedos de voz, isto é, uma mensagem gravada que pode ser transmitida ou reproduzida caso, por exemplo, clientes entrem em contato com a empresa e não tenha ninguém disponível para atendê-lo naquele momento.

API para SMS é uma ótima ferramenta de divulgação de lançamentos, promoções, eventos, serviços, etc.

Caso não saiba, **software** trata-se de uma sequência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado ou acontecimento.

Q2) Antes de comentar sobre os tipos de arquivos, é importante apresentar o que é uma API remota. As APIs remotas foram projetadas para interagir por meio de uma rede de comunicações. Quando falamos remota, queremos dizer que os recursos utilizados pela API estão em algum lugar fora do computador que realiza a solicitação. Como a rede de comunicações mais usada é a Internet, a maioria das APIs são projetadas com base em padrões da web. Nem todas as APIs remotas são web, mas é justo afirmar que, em geral, as APIs web são remotas.

As APIs web normalmente usam o protocolo HTTP para mensagens de solicitação e fornecem uma definição da estrutura das mensagens de resposta. Essas mensagens de resposta geralmente têm o formato de arquivo XML ou JSON. Tanto XML (Extensible Markup Language) quanto JSON (Javascript Object Notation) são formatos de preferência porque apresentam os dados de forma simplificada, facilitando a manipulação por outras aplicações.

Q3) Primeiro, REST (Representational State Transfer ou transferência de estado representacional) é um estilo de arquitetura. Por sua definição, trata-se de um conjunto de princípios e definições necessários para a criação de um projeto com interfaces bem definidas. APIs web que adotam as restrições de arquitetura da REST são chamadas de APIs RESTful. Uma API é considerada API RESTful se estiver em conformidade com seis restrições de arquitetura:

**Arquitetura cliente servidor** - a arquitetura REST é composta por clientes, servidores e recursos. Ela lida com as solicitações via HTTP.

**Sem monitoramento de estado (comunicação stateless)** - nenhum conteúdo do cliente é armazenado no servidor entre as solicitações. Em vez disso, as informações sobre o estado da sessão são mantidas com o cliente.

**Capacidade de cache** - o armazenamento em cache pode eliminar a necessidade de algumas interações entre o cliente e o servidor.

**Sistemas em camadas** - as interações entre cliente e servidor podem ser mediadas por camadas adicionais. Essas camadas podem oferecer recursos extras, como balanceamento de carga, caches compartilhados ou segurança.

**Código sob demanda (opcional)** - os servidores podem ampliar a funcionalidade de um cliente por meio da transferência de códigos executáveis.

**Interface uniforme** - essa restrição é essencial para o design de APIs RESTful e inclui quatro vertentes:

* **Identificação de recursos nas solicitações**: os recursos são identificados nas solicitações e separados das representações retornadas para o cliente.
* **Manipulação de recursos por meio de representações**: os clientes recebem arquivos que representam recursos. Essas representações precisam ter informações suficientes para permitir a modificação ou exclusão.
* **Mensagens autodescritivas**: cada mensagem retornada para um cliente contém informações suficientes para descrever como ele deve processá-las.
* **Hipermídia como plataforma do estado das aplicações**: depois de acessar um recurso, o cliente REST pode descobrir todas as outras ações disponíveis no momento por meio de hiperlinks.

Q4)Não exatamente.Só faz sentido saber o que é REST, já que RESTful é apenas a capacidade de fazer REST, ou seja, é uma questão gramatical.

Uma API RESTful é aquela que está em conformidade com os critérios estabelecidos pela **Transferência de Estado Representacional (REST)**. Assim, quando uma API é do tipo RESTful, significa que esse sistema consegue aplicar os princípios propostos por Roy Fielding em sua tese REST.

Como já definimos anteriormente, REST é um conjunto de princípios e restrições de arquitetura de softwares.

Uma API RESTful é aquela que está em conformidade com os critérios estabelecidos pela Transferência de Estado Representacional (REST).

Assim, quando uma API é do tipo RESTful, significa que esse sistema consegue aplicar os princípios propostos por Roy Fielding em sua tese REST.

A função básica de uma API RESTful é a mesma de navegar na internet. O cliente entra em contato com o servidor usando a API quando requer um recurso. Os desenvolvedores de API explicam como o cliente deve usar API REST na documentação da API da aplicação do servidor. Estas são as etapas gerais para qualquer chamada de API REST:

1. O cliente envia uma solicitação ao servidor. O cliente segue a documentação da API para formatar a solicitação de modo que o servidor entenda.
2. O servidor autentica o cliente e confirma que o cliente tem o direito de fazer essa solicitação.
3. O servidor recebe a solicitação e a processa internamente.
4. O servidor retorna uma resposta ao cliente. A resposta contém informações que indicam ao cliente se a solicitação foi bem-sucedida. A resposta também inclui informações solicitadas pelo cliente.

Q5) A solicitação do cliente da API RESTful pode conter:

**Identificador de recurso exclusivo -** O servidor identifica cada recurso com identificadores de recursos exclusivos. Para serviços REST, o servidor normalmente realiza a identificação de recursos usando um **uniform resource locator** (URL – localizador de recurso uniforme, em português). O URL especifica o caminho para o recurso. Um URL é semelhante ao endereço do site que você digita no navegador para visitar qualquer página da Web. O URL também é chamado de endpoint de solicitação e especifica claramente ao servidor o que o cliente requer.

**Método -** Os desenvolvedores geralmente implementam APIs RESTful usando o Hypertext Transfer Protocol (HTTP – Protocolo de Transferência de Hipertexto, em português). Um método HTTP informa ao servidor o que ele precisa fazer com o recurso. A seguir, estão quatro métodos HTTP comuns:

GET - Os clientes usam GET para acessar recursos localizados no URL especificado no servidor. Eles podem armazenar em cache solicitações do tipo GET e enviar parâmetros na solicitação da API RESTful para instruir o servidor a filtrar dados antes de enviar.

POST - Os clientes usam POST para enviar dados ao servidor. Eles incluem a representação de dados com a solicitação. Se enviarem a mesma solicitação POST várias vezes, criarão o mesmo recurso várias vezes.

PUT - Os clientes usam PUT para atualizar recursos existentes no servidor. Ao contrário do POST, o envio da mesma solicitação PUT várias vezes em um serviço da Web RESTful tem o mesmo resultado.

DELETE - Os clientes usam a solicitação DELETE para remover o recurso. Uma solicitação DELETE pode alterar o estado do servidor. No entanto, se o usuário não tiver a autenticação apropriada, a solicitação falhará.

**Cabeçalhos HTTP -** Os cabeçalhos de solicitação são os metadados trocados entre o cliente e o servidor. Por exemplo, o cabeçalho da solicitação indica o formato da solicitação e da resposta, fornece informações sobre o status da solicitação e assim por diante.

Dados - As solicitações da API REST podem incluir dados para POST, PUT e outros métodos HTTP para funcionarem com êxito.

Parâmetros - As solicitações da API RESTful podem incluir parâmetros que fornecem ao servidor mais detalhes sobre o que precisa ser feito. A seguir, estão alguns tipos diferentes de parâmetro.

* Parâmetros de caminho que especificam detalhes do URL.
* Parâmetros de consulta que solicitam mais informações sobre o recurso.
* Parâmetros de cookies que autenticam clientes rapidamente.

Q6) Segundo os princípios da REST (Representational State Transfer ou transferência de estado representacional) pode exigir que a resposta do servidor contenha os seguintes componentes principais:

**Linha de status** - A linha de status contém um código de status de três dígitos que comunica o sucesso ou a falha da solicitação. Por exemplo, os códigos 2XX indicam êxito, mas os códigos 4XX e 5XX indicam erros. Os códigos 3XX indicam redirecionamento de URL**.**

A seguir, estão alguns códigos de status comuns.

* 200: resposta genérica de êxito
* 201: resposta de êxito do método POST
* 400: solicitação incorreta que o servidor não pode processar
* 404: recurso não encontrado

**Corpo da mensagem** - O corpo da resposta contém a representação do recurso. O servidor seleciona um formato de representação apropriado com base no que os cabeçalhos da solicitação contêm. Os clientes podem solicitar informações nos formatos XML ou JSON, que definem como os dados são escritos em texto simples. Por exemplo, se o cliente solicitar o nome e a idade de uma pessoa chamada Carlos, o servidor retornará uma representação JSON da seguinte forma:

'{"name":"Carlos", "age":18}'

**Cabeçalhos** - A resposta também contém cabeçalhos ou metadados sobre a resposta. Elas fornecem mais contexto sobre a resposta e incluem informações como servidor, codificação, data e tipo de conteúdo.

Q7) Um serviço da Web RESTful deve autenticar solicitações antes de enviar uma resposta. **Autenticação é o processo de verificação de uma identidade**. Por exemplo, você pode provar sua identidade mostrando uma carteira de identidade ou carteira de motorista. Da mesma forma, os clientes do serviço RESTful devem provar a identidade ao servidor para estabelecer confiança.

A API RESTful tem quatro métodos de autenticação comuns:

**Autenticação de HTTP**

O HTTP define alguns esquemas de autenticação que você pode usar diretamente ao implementar a API REST. Estes são dois desses esquemas:

***Autenticação básica***

Na autenticação básica, o cliente envia o nome de usuário e a senha no cabeçalho da solicitação. Ele os codifica com base64, que é uma técnica de codificação que converte o par em um conjunto de 64 caracteres para transmissão segura.

***Autenticação do portador***

O termo autenticação do portador se refere ao processo de dar controle de acesso ao portador do token. O token do portador é normalmente uma cadeia de caracteres criptografada que o servidor gera em resposta a uma solicitação de login. O cliente envia o token nos cabeçalhos de solicitação para acessar recursos.

### Chaves de API

As chaves de API são outra opção para autenticação da API REST. Nessa abordagem, o servidor atribui um valor gerado exclusivo a um cliente iniciante. Sempre que o cliente tenta acessar recursos, ele usa a chave de API exclusiva para verificar a si mesmo. As chaves de API são menos seguras porque o cliente precisa transmitir a chave, o que a torna vulnerável a roubo de rede.

### OAuth

OAuth combina senhas e tokens para acesso de login altamente seguro a qualquer sistema. Primeiro, o servidor solicita uma senha e, depois, um token adicional para concluir o processo de autorização. Ele pode verificar o token a qualquer momento e também ao longo do tempo com escopo e longevidade específicos.

Q8) O padrão REST é o método de preferência na integração de aplicações e na conexão de componentes de softwares. Grandes sites como Twitter, Amazon e Linkedin utilizam APIs do tipo RESTful. E a tendência é que a computação em nuvem confirme cada vez mais a arquitetura API REST como via de regra no desenvolvimento de aplicações no futuro.

Q9)