Programação para Internet

Módulo 10

Introdução à Web Services com Java e Spring Boot

Prof. Dr. Daniel A. Furtado - FACOM/UFU

Conteúdo

- Conceito de web service
- Tipos de web services: SOAP x REST
- Métodos HTTP e conceito de Idempotência
- Introdução ao framework Spring e tecnologias associadas
- Passo a passo para criação de serviço de exemplo utilizando o Spring Boot

Web Services e APIs

- Um web service é uma entidade de software independente de linguagem, baseada em padrões, que aceita requisições HTTP especialmente formatadas de outras entidades de software em máquinas remotas, e produz respostas específicas da aplicação;
- Web services utilizam padrões de comunicação universais, como o protocolo HTTP, JSON, XML etc. para prover interoperabilidade;
- O ViaCep (viacep.com.br), utilizado anteriormente para busca de endereços a partir de CEPs, é um exemplo de web service;
- Em geral, os web services constituem APIs (*Application Programming Interface* - Interface de Programação de Aplicação), porém nem toda API é um web service (ex. API's do Windows em aplicações locais).

Tipos de Web Services

- Os web services são comumente categorizados de acordo com a tecnologia em que se baseiam, que são:
  - SOAP
  - REST

Tipos de Web Services

SOAP

- Acrônimo para **S**imple **O**bject **A**ccess **P**rotocol;
- Utiliza a **W**eb **S**ervices **D**escription **L**anguage (WSDL), uma linguagem baseada na XML, para a troca de mensagens entre aplicações;
- Como é baseado na XML, é independente de plataforma ou linguagem;
- Provê uma espécie de “envelope” para o envio de mensagens de serviços web através da Internet/Intranet;
- Mais comumente utilizado em APIs privadas, com medidas de segurança mais rígidas;
- O servidor geralmente mantém o estado, armazenando as mensagens anteriores trocadas com um cliente.

SOAP - Exemplo

1) Exemplo de mensagem de requisição SOAP para buscar no servidor a cotação de uma ação na bolsa

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>

</soap:Envelope>
```

Uma desvantagem clara do SOAP é o overhead com *metadados*, o que demanda uma maior largura de banda

2) Exemplo de uma mensagem de resposta SOAP com o preço da ação

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>

<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPriceResponse>
    <m:Price>34.5</m:Price>
  </m:GetStockPriceResponse>
</soap:Body>

</soap:Envelope>
```

Adaptado de *w3schools.com*

Tipos de Web Services

REST

- **Re**presentational **S**tate **T**ransfer;
- Estilo arquitetural para comunicação entre aplicações na Web;
- Se baseia nos métodos HTTP (GET, POST, PUT, DELETE), códigos de status e identificação dos recursos pela própria URL (endpoints);
- Diferente do SOAP, não impõe restrições ao formato da mensagem, mas apenas no comportamento das entidades envolvidas:
  - Mais flexível: o desenvolvedor pode utilizar o formato que for mais apropriado, como XML, JSON, texto etc.
- **Stateless**: toda requisição deve conter todas as informações necessárias para que o servidor possa processá-la;
- RESTful é o termo comumente utilizado para designar web services / APIs baseados no estilo REST.

Web Services e Idempotência

Um requisição HTTP é dita idempotente quando mantem a seguinte propriedade:

- Executar a requisição múltiplas vezes tem o mesmo efeito (no estado do servidor) que executá-la uma única vez;
- Em outras palavras, isso significa que executar a mesma requisição pela segunda ou terceira vez não irá alterar o estado da aplicação no servidor.

Principais Métodos HTTP para serviços RESTful

GET

- De acordo com especificação, GET deve ser utilizado para acesso (leitura) de recursos no servidor, mas não para alteração;
- Em caso de sucesso, retorna-se uma representação do recurso em formatos como JSON, XML, texto, etc. e o código de status 200 - Ok;
- Por definição, é idempotente. Isto significa que um usuário pode executar a mesma requisição várias vezes sem se preocupar em produzir efeitos adversos no servidor

POST

- Utilizado para criar novos recursos;
- Em geral, altera o estado da aplicação no servidor;
- Em caso de sucesso, deve-se retornar o código de status 201 – Created;
- Não é idempotente. Isto significa que sucessivas requisições idênticas utilizando o POST podem ter efeitos diferentes (por exemplo, podem resultar na criação de dois ou mais recursos contendo a mesma informação).

Principais Métodos HTTP para serviços RESTful

PUT

- Utilizado para operações de atualização/substituição por inteiro de um recurso no servidor;
- Semelhante ao método POST, porém com a diferença de ser idempotente;
- Em outras palavras, se um recurso é substituído por meio de uma requisição PUT e, na sequência, a mesma requisição é repetida outras vezes, então todas devem ter o mesmo resultado, uma vez que o recurso atualizado permanecerá no mesmo estado que estava logo após a primeira requisição.

DELETE

- Utilizado para excluir um recurso no servidor;
- Em caso de sucesso, deve-se retornar o código HTTP 200 (OK);
- É idempotente. Repetidas requisições para remoção do mesmo recurso devem ter o mesmo resultado (200 – OK), ou seja, o recurso foi apagado e continua apagado.

Endpoints e Rotas

- No contexto de web services, endpoints representam URLs específicas que os desenvolvedores devem utilizar para enviar solicitações e receber respostas (ex. viacep.com.br/ws/38400-100/json)
- Os endpoints pode ser combinados com os métodos HTTP para obtenção de diferentes rotas
  - GET minha.api.com/v1/book/1
  - PUT minha.api.com/v1/book/1
  - DELETE minha.api.com/v1/book/1

Métodos Suportados em Formulários HTML

- Vale destacar que, para submissão de formulários HTML utilizando a tag <form>, os únicos métodos suportados são GET e POST (não se deve utilizar, por exemplo, `<form method='put'>`);
- Entretanto, os demais métodos podem ser utilizados por meio de requisições HTTP utilizando o XMLHttpRequest ou a API Fetch.

Introdução ao Framework Spring

- **Spring** é um framework de código aberto amplamente utilizado no desenvolvimento de aplicações Java;
- É organizado em vários módulos e containers que permitem o desenvolvimento de vários tipos de aplicações, incluindo aplicações Web;
- Por exemplo, há módulos (**Core**, **Beans**) que implementam recursos fundamentais do framework como suporte a **Inversão de Controle** (IoC);
- Por outro lado, o módulo **Web-Servlet** dá suporte à arquitetura MVC (model-view-controller) para aplicações web.

Spring Boot

- Lançado em 2014, a ferramenta Spring Boot permite criar rapidamente um novo projeto utilizando o Spring
  - Vários recursos já estão pré-configurados, evitando a configuração detalhada de arquivos XML (o que é necessário quando se utiliza o Spring puro, sem o Spring Boot);
  - Simplifica a inclusão de dependências do projeto.

Ferramenta de Desenvolvimento

IntelliJ IDEA Ultimate

- Ambiente de desenvolvimento integrado (IDE) multiplataforma para linguagens baseadas na Java Virtual Machine (JVM) como Java e Kotlin;
- Também suporta outras linguagens por meio de plugins.

IntelliJ IDEA Community Edition

- Versão gratuita do IntelliJ IDEA, com recursos básicos;
- https://www.jetbrains.com/idea/download

Algumas Tecnologias Envolvidas

Apache Tomcat

- Servidor Web de código aberto desenvolvido pela Apache Software Foundation que implementa uma série de especificações para desenvolvimento de aplicações Web com Java.

Ferramenta de Automação de Compilação (build automation)

- Ferramenta para automatizar o processo de compilação (build) de software, gerenciando as dependências do projeto tais como bibliotecas e frameworks utilizados;
- Geralmente fazem o download automático das dependência do projeto e as mantêm em cache localmente;
- Alguns exemplos são o Gradle e o Maven.

Algumas Tecnologias Envolvidas

Gradle

- Ferramenta de automação de compilação de código aberto que utiliza scripts declarativos em Groovy ou Kotlin;
- Pode ser utilizada no desenvolvimento com Java e outras linguagens, mas sua popularidade é maior no desenvolvimento para Android;
- Suas vantagens incluem flexibilidade e desempenho.

Maven

- Ferramentas de automação de compilação desenvolvida pela Apache;
- Bastante utilizada no desenvolvimento com Java;
- Suas vantagens incluem maturidade, ampla documentação e ecosistema de plugins.
- Utiliza um arquivo de configuração em XML (pom.xml)

Exemplo de arquivo pom.xml utilizado no Maven

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https:/
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.0</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>demo</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
```

Criação de um serviço de exemplo com o Spring Boot

https://furtado.prof.ufu.br/site/teaching/PPI/Atividade-Intro-Spring.pdf