

Exercícios Desenvolvimento de Web Services

Estes exercícios tem como objetivo guiá-lo sobre os conhecimentos que julgamos necessário para que você consiga desenvolver aplicações que utilizam Web Services de maneira adequada.

Neste, estabelecemos os conhecimentos teóricos e práticos necessários para que você consiga desenvolver seus Web Services sem maiores problemas.

1 - Crie um projeto Maven. Você deve utilizar a opção “Create simple project”.

2 – Configure seu pom.xml conforme abaixo:

LEMBRE-SE DE ALTERAR O NOME DO GROUPID E ARTIFACTID CONFORME O NOME DO SEU PROJETO.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>
    <groupId>br.com.unitri.posjava.oauthresource</groupId>
    <artifactId>oauthresource</artifactId>
    <version>0.0.1-SNAPSHOT</version>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>1.4.0.RELEASE</version>
        <relativePath /> <!-- lookup parent from repository -->
    </parent>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
        <java.version>1.8</java.version>
        <spring.version>4.3.2.RELEASE</spring.version>
    </properties>

    <dependencies>

        <!-- Spring Boot dependencies -->

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <optional>true</optional>
        </dependency>

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

    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

```
        </plugin>
    </plugins>
</build>
</project>
```

A configuração deste pom lhe permitirá criar um Web Service Rest.

2 – Baseado no que foi apresentado durante a disciplina, o que você entende como Web Service?

3 – O que você entende como Web Service REST? Qual a diferença entre os Web Services REST do ponto de vista teórico e do ponto de vista do que o mercado tem praticado? Qual sua opinião sobre isto?

4 - Projete um Web Service Rest que atenda as necessidades abaixo:

- Criar um cliente
- Remover um cliente
- Buscar um cliente
- Atualizar um cliente

5 – O que são representações e recursos em Web Services REST?

6 - Permita que seu Web Service retorne representações no format JSON e XML. Por padrão, o Spring já lhe permite a formatação de representações em format JSON. Adicione a dependência abaixo, e faça o teste no seu cliente HTTP, verificando se agora seu Web Service retorna representações em formato XML.

```
<dependency>
    <groupId>com.fasterxml.jackson.dataformat</groupId>
    <artifactId>jackson-dataformat-xml</artifactId>
</dependency>
```

Lembrando que para fazer o teste você deve informar no seu cliente HTTP o header ACCEPT: application/xml

7 – Responda qual o papel de cada um dos códigos de retorno do HTTP abaixo:

1XX -
2XX -
3XX -
4XX -
5XX -

8 – Baseado nisso, verifique se o código desenvolvido na questão número 4 está retornando os códigos de forma correta.

9 – Descreva os principais métodos HTTP, tais como GET, POST, PUT, DELETE, OPTIONS.

10 – Verifique se o código desenvolvido na questão número 4 está retornando os códigos de forma correta.

11 – O modelo REST tem como princípio 6 constraints, as mesmas são listadas abaixo, escreva o papel de cada uma e como você acredita que as mesmas podem contribuir no desenvolvimento de

web services REST.

- Cliente / Servidor
- Stateless
- Caching
- Sistema em camadas
- Código sob-demanda
- Interface uniforme

12 – Adicione suporte a cache na sua API. Detalhes de como isso pode ser implementado, pode ser encontrado nos slides desta disciplina.

13 – Teoricamente, um Web Service só pode ser considerado REST quando de fato adicionamos suporte HATEOAS a ele. Descreva o que é HATEOAS e diga na sua opinião porque este recurso é interessante (ou não) no desenvolvimento de Web Services.

14 – Adicione suporte a HATEOAS no Web Service desenvolvido na questão número 4. Lembrando que para isso, você deve adicionar a dependência abaixo:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-hateoas</artifactId>
</dependency>
```

15 – O desenvolvimento de Web Services pode se dar por diversas formas, neste curso temos trabalhado com o framework Spring, baseado nisso responda:

a) De forma geral, como funciona o Front-Controller? Qual o papel dele no desenvolvimento de Web Services?

b) Como a anotação **@RequestMapping** e os atributos, **method** e **produces**, são utilizados para selecionar o método adequado para tratar uma determinada requisição?

16 – Segurança é um tópico importante no desenvolvimento de qualquer aplicação, para web services Rest, o framework OAuth 2 tem sido a escolha padrão, baseado nisso responda:

a) De maneira geral o que é o OAuth?

b) Pesquise na Web alguns serviços que o utilizam

c) Descreva os papéis apresentados pelo framework:

- Client
- Resource Server
- Authorization Server
- Resource Owner

d) O que são grant types? Descreva o fluxo dos grant types: Authorization Code, Password e Client Credentials.

17 – Analise os projetos abaixo no no repositório Github da disciplina:

- exampleclient-oauth2
- oauth2-authorizationserver
- oauth2-resourceserver

a) Execute os projetos em seu ambiente

b) Tente obter um token utilizando cada um dos grant-types:

- Password
- Client Credentials
- Authorization Code

c) Acesse o resource presente no projeto exampleclient-oauth2 utilizando o token obtido

18 – Assegure o seu projeto utilizando o Authorization Server.

19 – Faça testes no seu projeto.

20 – Sobre sua percepção geral sobre Web Services REST, responda:

a) Qual sua visão geral sobre Web Services REST?

b) Você acredita que o Spring é um framework adequado para o desenvolvimento de Web Services REST? Justifique.

c) Você já tem algum problema em mente que um Web Service REST pode lhe ajudar a resolver?