Spring Web Service

Spring WS - First Application

## Chúng ta hãy bắt đầu viết một SOAP thực tế dựa trên dịch vụ web với Spring-WS Framework. Trước khi chúng ta bắt đầu viết ví dụ đầu tiên sử dụng Spring-WS framework, phải đảm bảo rằng môi trường Spring-WS được thiết lập đúng như trong phần Spring Web Services - Environment Setup chapter. Chúng tôi giả định rằng các độc giả có một số kiến thức làm việc cơ bản với IDE Eclipse.

## Do đó, chúng ta hãy tiến hành viết một Ứng dụng Spring WS đơn giản hiển thị một phương thức dịch vụ web để book a leave trong HR Portal.

## Contract-first Approach

Spring-WS sử dụng cách tiếp Contract-first, có nghĩa là chúng ta nên chuẩn bị sẵn Cấu trúc XML trước khi viết bất kỳ code triển khai dựa trên JAVA nào. Chúng ta sẽ thử xác định LeaveRequest Object, với các object phụ - Leave và Employee.

Sau đây là các cấu trúc XML−

### Leave.xml

<Leave xmlns = "http://tutorialspoint.com/hr/schemas">

<StartDate>2016-07-03</StartDate>

<EndDate>2016-07-07</EndDate>

</Leave>

### Employee.xml

<Employee xmlns = "http://tutorialspoint.com/hr/schemas">

<Number>404</Number>

<FirstName>Mahesh</FirstName>

<LastName>Parashar</LastName>

</Employee>

### LeaveRequest.xml

<LeaveRequest xmlns = "http://tutorialspoint.com/hr/schemas">

<Leave>

<StartDate>2016-07-03</StartDate>

<EndDate>2016-07-07</EndDate>

</Leave>

<Employee>

<Number>404</Number>

<FirstName>Mahesh</FirstName>

<LastName>Parashar</LastName>

</Employee>

</LeaveRequest>

### hr.xsd

<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"

xmlns:hr = "http://tutorialspoint.com/hr/schemas"

elementFormDefault = "qualified"

targetNamespace = "http://tutorialspoint.com/hr/schemas">

<xs:element name = "LeaveRequest">

<xs:complexType>

<xs:all>

<xs:element name = "Leave" type = "hr:LeaveType"/>

<xs:element name = "Employee" type = "hr:EmployeeType"/>

</xs:all>

</xs:complexType>

</xs:element>

<xs:complexType name = "LeaveType">

<xs:sequence>

<xs:element name = "StartDate" type = "xs:date"/>

<xs:element name = "EndDate" type = "xs:date"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name = "EmployeeType">

<xs:sequence>

<xs:element name = "Number" type = "xs:integer"/>

<xs:element name = "FirstName" type = "xs:string"/>

<xs:element name = "LastName" type = "xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

## Create the Project

Bây giờ chúng ta hãy mở một command console , vào thư mục C: \ MVN và thực hiện lệnh mvn sau.

C:\MVN>mvn archetype:generate -DarchetypeGroupId = org.springframework.ws

-DarchetypeArtifactId = spring-ws-archetype -DgroupId = com.tutorialspoint.hr

-DartifactId = leaveService

Maven sẽ bắt đầu xử lý và sẽ tạo ra Java Application Project Structure.

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building Maven Stub Project (No POM) 1

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] Using property: groupId = com.tutorialspoint.hr

[INFO] Using property: artifactId = leaveService

Define value for property 'version': 1.0-SNAPSHOT: :

[INFO] Using property: package = com.tutorialspoint.hr

Confirm properties configuration:

groupId: com.tutorialspoint.hr

artifactId: leaveService

version: 1.0-SNAPSHOT

package: com.tutorialspoint.hr

Y: :

[INFO] -------------------------------------------------------------------------

---

[INFO] Using following parameters for creating project from Old (1.x) Archetype:

spring-ws-archetype:2.0.0-M1

[INFO] -------------------------------------------------------------------------

---

[INFO] Parameter: groupId, Value: com.tutorialspoint.hr

[INFO] Parameter: packageName, Value: com.tutorialspoint.hr

[INFO] Parameter: package, Value: com.tutorialspoint.hr

[INFO] Parameter: artifactId, Value: leaveService

[INFO] Parameter: basedir, Value: C:\mvn

[INFO] Parameter: version, Value: 1.0-SNAPSHOT

[INFO] project created from Old (1.x) Archetype in dir: C:\mvn\leaveService

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 35.989 s

[INFO] Finished at: 2017-01-21T11:18:31+05:30

[INFO] Final Memory: 17M/178M

[INFO] ------------------------------------------------------------------------

### Bây giờ hãy vào thư mục C: / MVN. Chúng ta sẽ thấy một dự án ứng dụng java được tạo có tên leftService (như được chỉ định trong artifactId). Cập nhật tệp pom.xml và thêm thư mục HumanResourceService.java và HumanResourceServiceImpl.java vào trong thư mục sau - C: \ MVN \ leftService \ src \ main \ java \ com \ guidespoint \ hr \ service. Khi đã xong, sau đó thêm AwayEndpoint.java trong thư mục sau - C: \ MVN \ leftService \ src \ main \ java \ com \ guidespoint \ hr \ ws.

### pom.xml

<?xml version = "1.0" encoding = "UTF-8"?>

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.tutorialspoint.hr</groupId>

<artifactId>leaveService</artifactId>

<packaging>war</packaging>

<version>1.0-SNAPSHOT</version>

<name>leaveService Spring-WS Application</name>

<url>http://www.springframework.org/spring-ws</url>

<build>

<finalName>leaveService</finalName>

</build>

<dependencies>

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-ws-core</artifactId>

<version>2.4.0.RELEASE</version>

</dependency>

<dependency>

<groupId>jdom</groupId>

<artifactId>jdom</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>jaxen</groupId>

<artifactId>jaxen</artifactId>

<version>1.1</version>

</dependency>

<dependency>

<groupId>wsdl4j</groupId>

<artifactId>wsdl4j</artifactId>

<version>1.6.2</version>

</dependency>

</dependencies>

</project>

### HumanResourceService.java

package com.tutorialspoint.hr.service;

import java.util.Date;

public interface HumanResourceService {

void bookLeave(Date startDate, Date endDate, String name);

}

### HumanResourceServiceImpl.java

package com.tutorialspoint.hr.service;

import java.util.Date;

import org.springframework.stereotype.Service;

@Service

public class HumanResourceServiceImpl implements HumanResourceService {

public void bookLeave(Date startDate, Date endDate, String name) {

System.out.println("Booking holiday for [" + startDate + "-" + endDate + "]

for [" + name + "] ");

}

}

### LeaveEndpoint.java

package com.tutorialspoint.hr.ws;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.ws.server.endpoint.annotation.Endpoint;

import org.springframework.ws.server.endpoint.annotation.PayloadRoot;

import org.springframework.ws.server.endpoint.annotation.RequestPayload;

import com.tutorialspoint.hr.service.HumanResourceService;

import org.jdom.Element;

import org.jdom.JDOMException;

import org.jdom.Namespace;

import org.jdom.xpath.XPath;

@Endpoint

public class LeaveEndpoint {

private static final String NAMESPACE\_URI = "http://tutorialspoint.com/hr/schemas";

private XPath startDateExpression;

private XPath endDateExpression;

private XPath nameExpression;

private HumanResourceService humanResourceService;

@Autowired

public LeaveEndpoint(HumanResourceService humanResourceService) throws JDOMException {

this.humanResourceService = humanResourceService;

Namespace namespace = Namespace.getNamespace("hr", NAMESPACE\_URI);

startDateExpression = XPath.newInstance("//hr:StartDate");

startDateExpression.addNamespace(namespace);

endDateExpression = XPath.newInstance("//hr:EndDate");

endDateExpression.addNamespace(namespace);

nameExpression = XPath.newInstance("concat(//hr:FirstName,' ',//hr:LastName)");

nameExpression.addNamespace(namespace);

}

@PayloadRoot(namespace = NAMESPACE\_URI, localPart = "LeaveRequest")

public void handleLeaveRequest(@RequestPayload Element leaveRequest) throws Exception {

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

Date startDate = dateFormat.parse(startDateExpression.valueOf(leaveRequest));

Date endDate = dateFormat.parse(endDateExpression.valueOf(leaveRequest));

String name = nameExpression.valueOf(leaveRequest);

humanResourceService.bookLeave(startDate, endDate, name);

}

}

### /WEB-INF/spring-ws-servlet.xml

<beans xmlns = "http://www.springframework.org/schema/beans"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xmlns:context = "http://www.springframework.org/schema/context"

xmlns:sws = "http://www.springframework.org/schema/web-services"

xsi:schemaLocation = "http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/web-services

http://www.springframework.org/schema/web-services/web-services-2.0.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:component-scan base-package = "com.tutorialspoint.hr"/>

<bean id = "humanResourceService"

class = "com.tutorialspoint.hr.service.HumanResourceServiceImpl" />

<sws:annotation-driven/>

<sws:dynamic-wsdl id = "leave"

portTypeName = "HumanResource"

locationUri = "/leaveService/"

targetNamespace = "http://tutorialspoint.com/hr/definitions">

<sws:xsd location = "/WEB-INF/hr.xsd"/>

</sws:dynamic-wsdl>

</beans>

### /WEB-INF/web.xml

<web-app xmlns = "http://java.sun.com/xml/ns/j2ee"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation = "http://java.sun.com/xml/ns/j2ee

http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"

version = "2.4">

<display-name>TutorialsPoint HR Leave Service</display-name>

<servlet>

<servlet-name>spring-ws</servlet-name>

<servlet-class>

org.springframework.ws.transport.http.MessageDispatcherServlet

</servlet-class>

<init-param>

<param-name>transformWsdlLocations</param-name>

<param-value>true</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>spring-ws</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

</web-app>

### /WEB-INF/hr.xsd

<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"

xmlns:hr = "http://tutorialspoint.com/hr/schemas"

elementFormDefault = "qualified"

targetNamespace = "http://tutorialspoint.com/hr/schemas">

<xs:element name = "LeaveRequest">

<xs:complexType>

<xs:all>

<xs:element name = "Leave" type = "hr:LeaveType"/>

<xs:element name = "Employee" type = "hr:EmployeeType"/>

</xs:all>

</xs:complexType>

</xs:element>

<xs:complexType name = "LeaveType">

<xs:sequence>

<xs:element name = "StartDate" type = "xs:date"/>

<xs:element name = "EndDate" type = "xs:date"/>

</xs:sequence>

</xs:complexType>

<xs:complexType name = "EmployeeType">

<xs:sequence>

<xs:element name = "Number" type = "xs:integer"/>

<xs:element name = "FirstName" type = "xs:string"/>

<xs:element name = "LastName" type = "xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:schema>

## Build the Project

Bây giờ chúng ta hãy mở command console , vào thư mục C: \ MVN \ leftService và thực hiện lệnh mvn sau.

C:\MVN\leaveService>mvn clean package

Maven sẽ bắt đầu xây dựng project.

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building leaveService Spring-WS Application 1.0-SNAPSHOT

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ leaveService ---

[INFO] Deleting C:\mvn\leaveService\target

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ leaveServi

ce ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] Copying 0 resource

[INFO]

[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ leaveService --

-

[INFO] Changes detected - recompiling the module!

[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. b

uild is platform dependent!

[INFO] Compiling 3 source files to C:\mvn\leaveService\target\classes

[INFO]

[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ le

aveService ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] skip non existing resourceDirectory C:\mvn\leaveService\src\test\resource

s

[INFO]

[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ leaveSe

rvice ---

[INFO] No sources to compile

[INFO]

[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ leaveService ---

[INFO] No tests to run.

[INFO]

[INFO] --- maven-war-plugin:2.2:war (default-war) @ leaveService ---

[INFO] Packaging webapp

[INFO] Assembling webapp [leaveService] in [C:\mvn\leaveService\target\leaveServ

ice]

[INFO] Processing war project

[INFO] Copying webapp resources [C:\mvn\leaveService\src\main\webapp]

[INFO] Webapp assembled in [7159 msecs]

[INFO] Building war: C:\mvn\leaveService\target\leaveService.war

[INFO] WEB-INF\web.xml already added, skipping

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 19.667 s

[INFO] Finished at: 2017-01-21T11:56:43+05:30

[INFO] Final Memory: 18M/173M

[INFO] ------------------------------------------------------------------------

### Import Project in Eclipse

## Thực hiện theo các bước được đưa ra dưới đây để nhập dự án trong Eclipse.

## • Mở Eclipse.

## • Chọn File → Import → option.

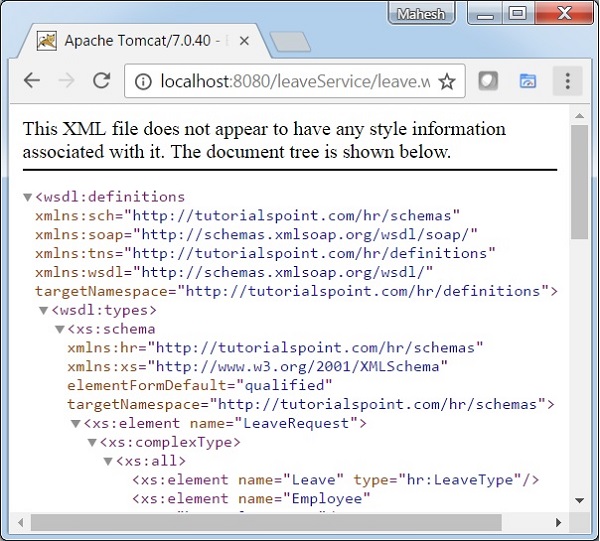
## • Chọn Maven Projects Option. Bấm vào nút tiếp theo.

## • Chọn Vị trí project, nơi project leftService được tạo bằng Maven.

## • Nhấp vào nút Finish.

## Run the Project

Khi chúng ta hoàn thành việc tạo tệp nguồn và cấu hình, xuất ứng dụng. Nhấp chuột phải vào ứng dụng, dùng tùy chọn Export → WAR File và lưu tệp leftService.war trong thư mục webapps của Tomcat. Khởi động máy chủ Tomcat và đảm bảo chúng ta có thể truy cập các trang web khác từ thư mục ứng dụng web bằng trình duyệt. Hãy thử truy cập URL - http: // localhost: 8080 / leftService / left.wsdl, nếu mọi thứ đều ổn với Ứng dụng Spring Web, chúng ta sẽ thấy màn hình sau.



# Spring WS - Static WSDL

Trong chương trước Spring -WS -First Application, chúng ta đã tạo WSDL một cách tự động bằng Spring WS Configuration. Trong trường hợp này, chúng ta sẽ thể hiện cách hiển thị WSDL hiện có bằng Spring WS.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Tạo một project với tên leftService trong gói com.tutorialspoint như đã được giải thích trong chương Spring WS - First Application. |
| 2 | Tạo một WSDL left.wsdl trong thư mục con / WEB-INF / wsdl. |
| 3 | Cập nhật spring-ws-servlet.xml trong thư mục con / WEB-INF. Chúng ta đang sử dụng thẻ static-wsdl ở đây thay vì Dynamic-wsdl. |
| 4 | Bước cuối cùng là tạo nội dung của tất cả các tệp nguồn và cấu hình và xuất ứng dụng như nội dung bên dưới. |

### /WEB-INF/spring-ws-servlet.xml

<wsdl:definitions xmlns:wsdl = "http://schemas.xmlsoap.org/wsdl/"

xmlns:soap = "http://schemas.xmlsoap.org/wsdl/soap/"

xmlns:schema = "http://tutorialspoint.com/hr/schemas"

xmlns:tns = "http://tutorialspoint.com/hr/definitions"

targetNamespace = "http://tutorialspoint.com/hr/definitions">

<wsdl:types>

<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">

<xsd:import namespace = "http://tutorialspoint.com/hr/schemas"

schemaLocation = "hr.xsd"/>

</xsd:schema>

</wsdl:types>

<wsdl:message name = "LeaveRequest">

<wsdl:part element = "schema:LeaveRequest" name = "LeaveRequest"/>

</wsdl:message>

<wsdl:portType name = "HumanResource">

<wsdl:operation name = "Leave">

<wsdl:input message = "tns:LeaveRequest" name = "LeaveRequest"/>

</wsdl:operation>

</wsdl:portType>

<wsdl:binding name = "HumanResourceBinding" type = "tns:HumanResource">

<soap:binding style = "document"

transport = "http://schemas.xmlsoap.org/soap/http"/>

<wsdl:operation name = "Leave">

<soap:operation soapAction = "http://mycompany.com/RequestLeave"/>

<wsdl:input name = "LeaveRequest">

<soap:body use = "literal"/>

</wsdl:input>

</wsdl:operation>

</wsdl:binding>

<wsdl:service name = "HumanResourceService">

<wsdl:port binding = "tns:HumanResourceBinding" name = "HumanResourcePort">

<soap:address location = "http://localhost:8080/leaveService/"/>

</wsdl:port>

</wsdl:service>

</wsdl:definitions>

### /WEB-INF/spring-ws-servlet.xml

<beans xmlns = "http://www.springframework.org/schema/beans"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xmlns:context = "http://www.springframework.org/schema/context"

xmlns:sws = "http://www.springframework.org/schema/web-services"

xsi:schemaLocation = "http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/web-services

http://www.springframework.org/schema/web-services/web-services-2.0.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:component-scan base-package = "com.tutorialspoint.hr"/>

<sws:annotation-driven/>

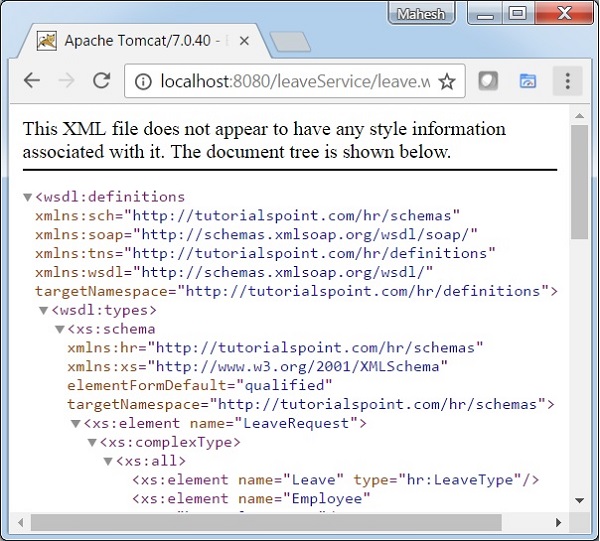
<sws:static-wsdl id = "leave" location = "/WEB-INF/wsdl/leave.wsdl"/>

</beans>

## Run the Project

Khi chúng ta hoàn thành việc tạo tệp nguồn và cấu hình, xuất ứng dụng. Nhấp chuột phải vào ứng dụng, dùng tùy chọn Export → WAR File và lưu tệp leftService.war trong thư mục webapps của Tomcat.

Khởi động máy chủ Tomcat và đảm bảo chúng ta có thể truy cập các trang web khác từ thư mục ứng dụng web bằng trình duyệt. Hãy thử truy cập URL - http: // localhost: 8080 / leftService / left.wsdl, nếu mọi thứ đều ổn với Ứng dụng Spring Web, chúng ta sẽ thấy màn hình sau.



# Spring WS - Writing Server

Trong chương này, chúng ta sẽ hiểu cách tạo một server ứng dụng web bằng Spring WS.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Tạo một dự án với tên countryService trong gói com.tutorialspoint như được giải thích trong chương Spring WS - First Application . |
| 2 | Tạo country.xsd, các lớp miền, CountryRep repository và CountryEndPoint như được giải thích trong các bước sau. |
| 3 | Cập nhật spring-ws-servlet.xml trong thư mục con / WEB-INF. |
| 4 | Bước cuối cùng là tạo nội dung cho tất cả các tệp nguồn và cấu hình và xuất ứng dụng như được giải thích bên dưới. |

### countries.xsd

<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema"

xmlns:tns = "http://tutorialspoint/schemas"

targetNamespace = "http://tutorialspoint/schemas"

elementFormDefault = "qualified">

<xs:element name = "getCountryRequest">

<xs:complexType>

<xs:sequence>

<xs:element name = "name" type = "xs:string"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:element name = "getCountryResponse">

<xs:complexType>

<xs:sequence>

<xs:element name = "country" type = "tns:country"/>

</xs:sequence>

</xs:complexType>

</xs:element>

<xs:complexType name = "country">

<xs:sequence>

<xs:element name = "name" type = "xs:string"/>

<xs:element name = "population" type = "xs:int"/>

<xs:element name = "capital" type = "xs:string"/>

<xs:element name = "currency" type = "tns:currency"/>

</xs:sequence>

</xs:complexType>

<xs:simpleType name = "currency">

<xs:restriction base = "xs:string">

<xs:enumeration value = "GBP"/>

<xs:enumeration value = "USD"/>

<xs:enumeration value = "INR"/>

</xs:restriction>

</xs:simpleType>

</xs:schema>

## Create the Project

Chúng ta hãy mở command console, vào thư mục C: \ MVN và thực hiện lệnh mvn sau.

C:\MVN>mvn archetype:generate -DarchetypeGroupId = org.springframework.ws

-DarchetypeArtifactId = spring-ws-archetype -DgroupId = com.tutorialspoint

-DartifactId = countryService

Maven sẽ bắt đầu xử lý và sẽ tạo ra Java Application Project Structure.

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building Maven Stub Project (No POM) 1

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] Using property: groupId = com.tutorialspoint

[INFO] Using property: artifactId = countryService

Define value for property 'version': 1.0-SNAPSHOT: :

[INFO] Using property: package = com.tutorialspoint

Confirm properties configuration:

groupId: com.tutorialspoint

artifactId: countryService

version: 1.0-SNAPSHOT

package: com.tutorialspoint

Y: :

[INFO] -------------------------------------------------------------------------

---

[INFO] Using following parameters for creating project from Old (1.x) Archetype:

spring-ws-archetype:2.0.0-M1

[INFO] -------------------------------------------------------------------------

---

[INFO] Parameter: groupId, Value: com.tutorialspoint

[INFO] Parameter: packageName, Value: com.tutorialspoint

[INFO] Parameter: package, Value: com.tutorialspoint

[INFO] Parameter: artifactId, Value: countryService

[INFO] Parameter: basedir, Value: C:\mvn

[INFO] Parameter: version, Value: 1.0-SNAPSHOT

[INFO] project created from Old (1.x) Archetype in dir: C:\mvn\countryService

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 35.989 s

[INFO] Finished at: 2017-01-21T11:18:31+05:30

[INFO] Final Memory: 17M/178M

[INFO] ------------------------------------------------------------------------

### Bây giờ hãy vào thư mục C: / MVN. Chúng ta sẽ thấy một project java application được tạo có tên countryService (như được chỉ định trong artifactId). Cập nhật tệp pom.xml.

### pom.xml

<?xml version = "1.0" encoding = "UTF-8"?>

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.tutorialspoint.hr</groupId>

<artifactId>countryService</artifactId>

<packaging>war</packaging>

<version>1.0-SNAPSHOT</version>

<name>countryService Spring-WS Application</name>

<url>http://www.springframework.org/spring-ws</url>

<build>

<finalName>countryService</finalName>

</build>

<dependencies>

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-ws-core</artifactId>

<version>2.4.0.RELEASE</version>

</dependency>

<dependency>

<groupId>jdom</groupId>

<artifactId>jdom</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>jaxen</groupId>

<artifactId>jaxen</artifactId>

<version>1.1</version>

</dependency>

<dependency>

<groupId>wsdl4j</groupId>

<artifactId>wsdl4j</artifactId>

<version>1.6.2</version>

</dependency>

</dependencies>

</project>

## Create Domain Classes

Sao chép countries.xsd trong thư mục C: \ mvn \ countryService \ src \ main \ resources. Chúng ta hãy mở command console, vào thư mục C: \ mvn \ countryService \ src \ main \ resource và thực hiện lệnh xjc sau để tạo các lớp miền bằng cách sử dụng country.xsd.

C:\MVN\countryService\src\main\resources>xjc -p com.tutorialspoint countries.xsd

Maven sẽ bắt đầu xử lý và sẽ tạo các lớp miền trong gói com.tutorialspoint.

parsing a schema...

compiling a schema...

com\tutorialspoint\Country.java

com\tutorialspoint\Currency.java

com\tutorialspoint\GetCountryRequest.java

com\tutorialspoint\GetCountryResponse.java

com\tutorialspoint\ObjectFactory.java

com\tutorialspoint\package-info.java

### Tạo thư mục java trong thư mục C: \ mvn \ countryService \ src \ main. Sao chép tất cả các lớp trong thư mục C: \ mvn \ countryService \ src \ main \ java. Tạo CountryRep repository và CountryEndPoint để thể hiện country database và country server tương ứng.

### CountryRepository.java

package com.tutorialspoint;

import java.util.ArrayList;

import java.util.List;

import org.springframework.beans.propertyeditors.CurrencyEditor;

import org.springframework.stereotype.Component;

import org.springframework.util.Assert;

@Component

public class CountryRepository {

private static final List<Country> countries = new ArrayList<Country>();

public CountryRepository(){

initData();

}

public void initData() {

Country us = new Country();

us.setName("United States");

us.setCapital("Washington");

us.setCurrency(Currency.USD);

us.setPopulation(46704314);

countries.add(us);

Country india = new Country();

india.setName("India");

india.setCapital("New Delhi");

india.setCurrency(Currency.INR);

india.setPopulation(138186860);

countries.add(india);

Country uk = new Country();

uk.setName("United Kingdom");

uk.setCapital("London");

uk.setCurrency(Currency.GBP);

uk.setPopulation(63705000);

countries.add(uk);

}

public Country findCountry(String name) {

Assert.notNull(name);

Country result = null;

for (Country country : countries) {

if (name.trim().equals(country.getName())) {

result = country;

}

}

return result;

}

}

### CountryEndPoint.java

package com.tutorialspoint.ws;

import org.jdom.JDOMException;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.ws.server.endpoint.annotation.Endpoint;

import org.springframework.ws.server.endpoint.annotation.PayloadRoot;

import org.springframework.ws.server.endpoint.annotation.RequestPayload;

import org.springframework.ws.server.endpoint.annotation.ResponsePayload;

import com.tutorialspoint.Country;

import com.tutorialspoint.CountryRepository;

import com.tutorialspoint.GetCountryRequest;

import com.tutorialspoint.GetCountryResponse;

@Endpoint

public class CountryEndPoint {

private static final String NAMESPACE\_URI = "http://tutorialspoint/schemas";

private CountryRepository countryRepository;

@Autowired

public CountryEndPoint(CountryRepository countryRepository) throws JDOMException {

this.countryRepository = countryRepository;

}

@PayloadRoot(namespace = NAMESPACE\_URI, localPart = "getCountryRequest")

@ResponsePayload

public GetCountryResponse getCountry(@RequestPayload GetCountryRequest request)

throws JDOMException {

Country country = countryRepository.findCountry(request.getName());

GetCountryResponse response = new GetCountryResponse();

response.setCountry(country);

return response;

}

}

### /WEB-INF/spring-ws-servlet.xml

<beans xmlns = "http://www.springframework.org/schema/beans"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xmlns:context = "http://www.springframework.org/schema/context"

xmlns:sws = "http://www.springframework.org/schema/web-services"

xsi:schemaLocation = "http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/web-services

http://www.springframework.org/schema/web-services/web-services-2.0.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:component-scan base-package = "com.tutorialspoint"/>

<sws:annotation-driven/>

<sws:dynamic-wsdl id="countries"

portTypeName = "CountriesPort"

locationUri = "/countryService/"

targetNamespace = "http://tutorialspoint.com/definitions">

<sws:xsd location = "/WEB-INF/countries.xsd"/>

</sws:dynamic-wsdl>

</beans>

### /WEB-INF/web.xml

<web-app xmlns = "http://java.sun.com/xml/ns/j2ee"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation = "http://java.sun.com/xml/ns/j2ee

http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd"

version = "2.4">

<display-name>TutorialsPoint Country Service</display-name>

<servlet>

<servlet-name>spring-ws</servlet-name>

<servlet-class>org.springframework.ws.transport.http.MessageDispatcherServlet

</servlet-class>

<init-param>

<param-name>transformWsdlLocations</param-name>

<param-value>true</param-value>

</init-param>

</servlet>

<servlet-mapping>

<servlet-name>spring-ws</servlet-name>

<url-pattern>/\*</url-pattern>

</servlet-mapping>

</web-app>

## Build the Project

Hãy mở command console . Chuyển đến thư mục C: \ MVN \ countryService và thực hiện lệnh mvn sau.

C:\MVN\countryService>mvn clean package

Maven sẽ bắt đầu xây dựng project.

INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building countryService Spring-WS Application 1.0-SNAPSHOT

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ countryService ---

[INFO] Deleting C:\mvn\countryService\target

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ countrySer

vice ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] Copying 0 resource

[INFO]

[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ countryService

---

[INFO] Changes detected - recompiling the module!

[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e.

build is platform dependent!

[INFO] Compiling 4 source files to C:\mvn\countryService\target\classes

[INFO]

[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ co

untryService ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] skip non existing resourceDirectory C:\mvn\countryService\src\test\resour

ces

[INFO]

[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ country

Service ---

[INFO] No sources to compile

[INFO]

[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ countryService ---

[INFO] No tests to run.

[INFO]

[INFO] --- maven-war-plugin:2.2:war (default-war) @ countryService ---

[INFO] Packaging webapp

[INFO] Assembling webapp [countryService] in [C:\mvn\countryService\target\count

ryService]

[INFO] Processing war project

[INFO] Copying webapp resources [C:\mvn\countryService\src\main\webapp]

[INFO] Webapp assembled in [5137 msecs]

[INFO] Building war: C:\mvn\countryService\target\countryService.war

[INFO] WEB-INF\web.xml already added, skipping

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 16.484 s

[INFO] Finished at: 2017-01-28T09:07:59+05:30

[INFO] Final Memory: 19M/170M

[INFO] ------------------------------------------------------------------------

## Run the Project

Sau khi đã tạo các tệp nguồn và cấu hình, hãy xuất tệp countryService.war trong thư mục ứng dụng web của Tomcat.

Bây giờ, hãy khởi động máy chủ Tomcat và đảm bảo nếu chúng ta có thể truy cập các trang web khác từ thư mục ứng dụng web bằng trình duyệt tiêu chuẩn. Tạo một yêu cầu POST cho URL - http: // localhost: 8080 / countryService / . Sử dụng bất kỳ SOAP client nào thực hiện yêu cầu sau-

<x:Envelope xmlns:x = "http://schemas.xmlsoap.org/soap/envelope/"

xmlns:tns = "http://tutorialspoint/schemas">

<x:Header/>

<x:Body>

<tns:getCountryRequest>

<tns:name>United States</tns:name>

</tns:getCountryRequest>

</x:Body>

</x:Envelope>

Bạn sẽ thấy kết quả sau đây.

<SOAP-ENV:Envelope xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/">

<SOAP-ENV:Header/>

<SOAP-ENV:Body>

<ns2:getCountryResponse xmlns:ns2 = "http://tutorialspoint/schemas">

<ns2:country>

<ns2:name>United States</ns2:name>

<ns2:population>46704314</ns2:population>

<ns2:capital>Washington</ns2:capital>

<ns2:currency>USD</ns2:currency>

</ns2:country>

</ns2:getCountryResponse>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

# Spring WS - Unit Test Server

Trong chương này, chúng ta sẽ hiểu cách kiểm tra đơn vị dịch vụ ứng dụng web được tạo bằng cách sử dụng Spring WS.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Cập nhật project countryService được tạo trong chương Spring WS - Write Server. Thêm thư mục src / test / java. |
| 2 | Tạo CustomerEndPointTest.java trong thư mục - src / test / java / com / guidespoint / ws và sau đó cập nhật POM.xml với chi tiết như bên dưới. |
| 3 | Thêm spring-context.xml trong thư mục con src / main / resource. |
| 4 | Bước cuối cùng là tạo nội dung cho tất cả các tệp nguồn và cấu hình và kiểm tra ứng dụng như được giải thích bên dưới. |

### POM.xml

<?xml version = "1.0" encoding = "UTF-8"?>

<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/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.tutorialspoint</groupId>

<artifactId>countryService</artifactId>

<packaging>war</packaging>

<version>1.0-SNAPSHOT</version>

<name>countryService Spring-WS Application</name>

<url>http://www.springframework.org/spring-ws</url>

<build>

<finalName>countryService</finalName>

</build>

<dependencies>

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-ws-core</artifactId>

<version>2.4.0.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>2.5</version>

</dependency>

<dependency>

<groupId>org.springframework.ws</groupId>

<artifactId>spring-ws-test</artifactId>

<version>2.4.0.RELEASE</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>3.1.2.RELEASE</version>

</dependency>

<dependency>

<groupId>jdom</groupId>

<artifactId>jdom</artifactId>

<version>1.0</version>

</dependency>

<dependency>

<groupId>jaxen</groupId>

<artifactId>jaxen</artifactId>

<version>1.1</version>

</dependency>

<dependency>

<groupId>wsdl4j</groupId>

<artifactId>wsdl4j</artifactId>

<version>1.6.2</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.5</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

### spring-context.xml

<beans xmlns = "http://www.springframework.org/schema/beans"

xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"

xmlns:context = "http://www.springframework.org/schema/context"

xmlns:sws = "http://www.springframework.org/schema/web-services"

xsi:schemaLocation = "http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/web-services

http://www.springframework.org/schema/web-services/web-services-2.0.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context-3.0.xsd">

<context:component-scan base-package = "com.tutorialspoint"/>

<sws:annotation-driven/>

<bean id = "schema" class = "org.springframework.core.io.ClassPathResource">

<constructor-arg index = "0" value = "countries.xsd" />

</bean>

</beans>

### CustomerEndPointTest.java

package com.tutorialspoint.ws;

import javax.xml.transform.Source;

import org.junit.Before;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.beans.factory.xml.XmlBeanDefinitionReader;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.GenericApplicationContext;

import org.springframework.test.context.ContextConfiguration;

import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import org.springframework.ws.test.server.MockWebServiceClient;

import org.springframework.xml.transform.StringSource;

import static org.springframework.ws.test.server.RequestCreators.withPayload;

import static org.springframework.ws.test.server.ResponseMatchers.payload;

@RunWith(SpringJUnit4ClassRunner.class)

@ContextConfiguration( locations = "/spring-context.xml" )

public class CustomerEndPointTest {

@Autowired

private ApplicationContext applicationContext;

private MockWebServiceClient mockClient;

@Before

public void createClient() {

mockClient = MockWebServiceClient.createClient(applicationContext);

GenericApplicationContext ctx = (GenericApplicationContext) applicationContext;

final XmlBeanDefinitionReader definitionReader = new XmlBeanDefinitionReader(ctx);

definitionReader.setValidationMode(XmlBeanDefinitionReader.VALIDATION\_NONE);

definitionReader.setNamespaceAware(true);

}

@Test

public void testCountryEndpoint() throws Exception {

Source requestPayload = new StringSource(

"<getCountryRequest xmlns = 'http://tutorialspoint/schemas'>"+

"<name>United States</name>"+

"</getCountryRequest>");

Source responsePayload = new StringSource(

"<getCountryResponse xmlns='http://tutorialspoint/schemas'>" +

"<country>" +

"<name>United States</name>"+

"<population>46704314</population>"+

"<capital>Washington</capital>"+

"<currency>USD</currency>"+

"</country>"+

"</getCountryResponse>");

mockClient.sendRequest(withPayload(requestPayload)).andExpect(payload(responsePayload));

}

}

## Build the Project

Hãy mở command console , vào thư mục C: \ MVN \ countryService và thực hiện lệnh mvn sau.

C:\MVN\countryService>mvn test

Maven sẽ bắt đầu xây dựng và thử nghiệm project.

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building countryService Spring-WS Application 1.0-SNAPSHOT

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ countrySer

vice ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] Copying 2 resources

[INFO]

[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ countryService

---

[INFO] Nothing to compile - all classes are up to date

[INFO]

[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ co

untryService ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] skip non existing resourceDirectory C:\MVN\countryService\src\test\resour

ces

[INFO]

[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ country

Service ---

[INFO] Nothing to compile - all classes are up to date

[INFO]

[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ countryService ---

[INFO] Surefire report directory: C:\MVN\countryService\target\surefire-reports

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.tutorialspoint.ws.CustomerEndPointTest

Feb 27, 2017 11:49:30 AM org.springframework.test.context.TestContextManager ret

rieveTestExecutionListeners

INFO: @TestExecutionListeners is not present for class [class com.tutorialspoint

.ws.CustomerEndPointTest]: using defaults.

Feb 27, 2017 11:49:30 AM org.springframework.beans.factory.xml.XmlBeanDefinition

Reader loadBeanDefinitions

INFO: Loading XML bean definitions from class path resource [spring-context.xml]

Feb 27, 2017 11:49:30 AM org.springframework.context.support.GenericApplicationC

ontext prepareRefresh

INFO: Refreshing org.springframework.context.support.GenericApplicationContext@b

2eddc0: startup date [Mon Feb 27 11:49:30 IST 2017]; root of context hierarchy

Feb 27, 2017 11:49:31 AM org.springframework.ws.soap.addressing.server.Annotatio

nActionEndpointMapping afterPropertiesSet

INFO: Supporting [WS-Addressing August 2004, WS-Addressing 1.0]

Feb 27, 2017 11:49:31 AM org.springframework.ws.soap.saaj.SaajSoapMessageFactory

afterPropertiesSet

INFO: Creating SAAJ 1.3 MessageFactory with SOAP 1.1 Protocol

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.386 sec

Feb 27, 2017 11:49:31 AM org.springframework.context.support.GenericApplicationC

ontext doClose

INFO: Closing org.springframework.context.support.GenericApplicationContext@b2ed

dc0: startup date [Mon Feb 27 11:49:30 IST 2017]; root of context hierarchy

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 3.517 s

[INFO] Finished at: 2017-02-27T11:49:31+05:30

[INFO] Final Memory: 11M/109M

[INFO] ------------------------------------------------------------------------

# Spring WS - Writing Client

Trong chương này, chúng ta sẽ tìm hiểu cách tạo một client cho server ứng dụng web (cách tạo trong chương Spring WS - Writing Server viết bằng Spring WS).

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Cập nhật project countryService trong gói com.tutorialspoint như được giải thích trong chương Spring WS - Writing Server. |
| 2 | Tạo CountryServiceClient.java trong gói com.tutorialspoint.client và MainApp.java trong gói com.tutorialspoint như được giải thích trong các bước sau. |

### CountryServiceClient.java

package com.tutorialspoint.client;

import org.springframework.ws.client.core.support.WebServiceGatewaySupport;

import com.tutorialspoint.GetCountryRequest;

import com.tutorialspoint.GetCountryResponse;

public class CountryServiceClient extends WebServiceGatewaySupport {

public GetCountryResponse getCountryDetails(String country){

String uri = "http://localhost:8080/countryService/";

GetCountryRequest request = new GetCountryRequest();

request.setName(country);

GetCountryResponse response =(GetCountryResponse) getWebServiceTemplate()

.marshalSendAndReceive(uri, request);

return response;

}

}

### MainApp.java

package com.tutorialspoint;

import org.springframework.oxm.jaxb.Jaxb2Marshaller;

import com.tutorialspoint.client.CountryServiceClient;

public class MainApp {

public static void main(String[] args) {

CountryServiceClient client = new CountryServiceClient();

Jaxb2Marshaller marshaller = new Jaxb2Marshaller();

marshaller.setContextPath("com.tutorialspoint");

client.setMarshaller(marshaller);

client.setUnmarshaller(marshaller);

GetCountryResponse response = client.getCountryDetails("United States");

System.out.println("Country : " + response.getCountry().getName());

System.out.println("Capital : " + response.getCountry().getCapital());

System.out.println("Population : " + response.getCountry().getPopulation());

System.out.println("Currency : " + response.getCountry().getCurrency());

}

}

## Start the Web Service

## Khởi động máy chủ Tomcat và đảm bảo rằng có thể truy cập các trang web khác nhau từ thư mục ứng dụng web bằng trình duyệt tiêu chuẩn.

## Test Web Service Client

Nhấp chuột phải vào MainApp.java trong ứng dụng của bạn trong Eclipse và sử dụng run as Java Application. Nếu mọi thứ đều ổn với ứng dụng, nó sẽ in thông báo sau.

Country : United States

Capital : Washington

Population : 46704314

Currency : USD

# Xong, chúng ta đã tạo một Client - CountryServiceClient.java cho SOAP dựa trên dịch vụ web . MainApp sử dụng CountryServiceClient để tác động đến dịch vụ web, thực hiện một yêu cầu bài đăng và nhận dữ liệu.

# Spring WS - Unit Test Client

Trong chương này, chúng ta sẽ tìm hiểu cách kiểm tra đơn vị máy khách được tạo trong Spring WS - Writing Client cho server ứng dụng web được tạo trong chương Spring WS - Writing Server viết bằng Spring WS.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Cập nhật project countryService trong gói com.tutorialspoint như được giải thích trong chương Spring WS - Writing Server. |
| 2 | Tạo CountryServiceClientTest.java trong gói com.tutorialspoint trong thư mục SRC → Test→ Java như được giải thích trong các bước được đưa ra dưới đây. |

### CountryServiceClientTest.java

package com.tutorialspoint;

import static org.junit.Assert.\*;

import org.junit.Assert;

import org.junit.Before;

import org.junit.Test;

import org.springframework.oxm.jaxb.Jaxb2Marshaller;

import com.tutorialspoint.client.CountryServiceClient;

public class CountryServiceClientTest {

CountryServiceClient client;

@Before

public void setUp() throws Exception {

client = new CountryServiceClient();

Jaxb2Marshaller marshaller = new Jaxb2Marshaller();

marshaller.setContextPath("com.tutorialspoint");

client.setMarshaller(marshaller);

client.setUnmarshaller(marshaller);

}

@Test

public void test() {

GetCountryResponse response = client.getCountryDetails("United States");

Country expectedCountry = new Country();

expectedCountry.setCapital("Washington");

Country actualCountry = response.getCountry();

Assert.assertEquals(expectedCountry.getCapital(), actualCountry.getCapital());

}

}

### Start the Web Service

## Khởi động máy chủ Tomcat và đảm bảo có thể truy cập các trang web khác từ thư mục webapps bằng trình duyệt tiêu chuẩn.

## Unit Test Web Service Client

Hãy mở command console , vào thư mục C: \ MVN \ countryService và thực hiện lệnh mvn sau.

C:\MVN\countryService>mvn test

Maven sẽ bắt đầu xây dựng và thử nghiệm project.

[INFO] Scanning for projects...

[INFO]

[INFO] ------------------------------------------------------------------------

[INFO] Building countryService Spring-WS Application 1.0-SNAPSHOT

[INFO] ------------------------------------------------------------------------

[INFO]

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ countrySer

vice ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] Copying 2 resources

[INFO]

[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ countryService

---

[INFO] Changes detected - recompiling the module!

[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. b

uild is platform dependent!

[INFO] Compiling 10 source files to C:\MVN\countryService\target\classes

[INFO]

[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ co

untryService ---

[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources,

i.e. build is platform dependent!

[INFO] skip non existing resourceDirectory C:\MVN\countryService\src\test\resour

ces

[INFO]

[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ country

Service ---

[INFO] Changes detected - recompiling the module!

[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. b

uild is platform dependent!

[INFO] Compiling 2 source files to C:\MVN\countryService\target\test-classes

[INFO]

[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ countryService ---

[INFO] Surefire report directory: C:\MVN\countryService\target\surefire-reports

-------------------------------------------------------

T E S T S

-------------------------------------------------------

Running com.tutorialspoint.CountryServiceClientTest

Feb 27, 2017 8:45:26 PM org.springframework.ws.soap.saaj.SaajSoapMessageFactory

afterPropertiesSet

INFO: Creating SAAJ 1.3 MessageFactory with SOAP 1.1 Protocol

Feb 27, 2017 8:45:26 PM org.springframework.oxm.jaxb.Jaxb2Marshaller createJaxbC

ontextFromContextPath

INFO: Creating JAXBContext with context path [com.tutorialspoint]

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.457 sec

Running com.tutorialspoint.ws.CustomerEndPointTest

Feb 27, 2017 8:45:27 PM org.springframework.test.context.TestContextManager retr

ieveTestExecutionListeners

INFO: @TestExecutionListeners is not present for class [class com.tutorialspoint

.ws.CustomerEndPointTest]: using defaults.

Feb 27, 2017 8:45:27 PM org.springframework.beans.factory.xml.XmlBeanDefinitionR

eader loadBeanDefinitions

INFO: Loading XML bean definitions from class path resource [spring-context.xml]

Feb 27, 2017 8:45:27 PM org.springframework.context.support.GenericApplicationCo

ntext prepareRefresh

INFO: Refreshing org.springframework.context.support.GenericApplicationContext@5

17c642: startup date [Mon Feb 27 20:45:27 IST 2017]; root of context hierarchy

Feb 27, 2017 8:45:28 PM org.springframework.ws.soap.addressing.server.Annotation

ActionEndpointMapping afterPropertiesSet

INFO: Supporting [WS-Addressing August 2004, WS-Addressing 1.0]

Feb 27, 2017 8:45:28 PM org.springframework.ws.soap.saaj.SaajSoapMessageFactory

afterPropertiesSet

INFO: Creating SAAJ 1.3 MessageFactory with SOAP 1.1 Protocol

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.243 sec

Feb 27, 2017 8:45:28 PM org.springframework.context.support.GenericApplicationCo

ntext doClose

INFO: Closing org.springframework.context.support.GenericApplicationContext@517c

642: startup date [Mon Feb 27 20:45:27 IST 2017]; root of context hierarchy

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 5.686 s

[INFO] Finished at: 2017-02-27T20:45:28+05:30

[INFO] Final Memory: 17M/173M

[INFO] ------------------------------------------------------------------------