

Web services

🏠 HOME (<http://javastudy.ru>) > Web services (<http://javastudy.ru/category/web-services/>)
> SOAP java веб сервис. Пример Hello World example

📅 22.06.2016 (<http://javastudy.ru/web-services/soap-java-hello-world-example/>) / обновлено : 20.09.2016 ✎ Nick V. Понравилось: 8

[Web services \(<http://javastudy.ru/category/web-services/>\)](http://javastudy.ru/category/web-services/)

SOAP java веб сервис. Пример Hello World example

Пример Java web-service с использованием технологии SOAP.

| Используемые технологии и библиотеки

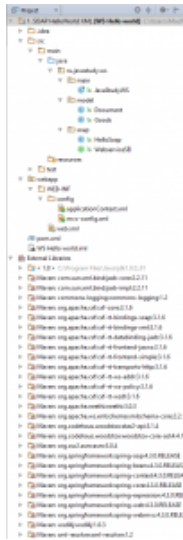
- Apache CXF 3.1.6
- Spring MVC 4.3.0.Release

1. Описание задачи

Создать простейший веб-сервис с использованием технологии SOAP. Написать простой класс веб сервиса на стороне сервера и обратиться к нему с помощью кода клиентской части.

2. Структура проекта





(<http://javastudy.ru/wp-content/uploads/2016/06/SOAPProjectStructure.png>)

Создан один класс описывающий SOAP веб-сервис (HelloSoap). Для демонстрации вывода Java объекта в xml формате используются два класса (Goods, Document). Для обращение к SOAP веб-сервису была написана простая клиентская часть, которая описана в классе JavaStudyWS.

3. pom.xml

Для реализации веб-сервиса используется библиотека Apache CXF.



```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <project xmlns="http://maven.apache.org/POM/4.0.0"
3      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4      xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.
5  0.0.xsd">
6      <modelVersion>4.0.0</modelVersion>
7
8      <groupId>ru.javastudy</groupId>
9      <artifactId>javastudy_webservices</artifactId>
10     <version>1.0</version>
11
12     <properties>
13
14         <java.version>1.8</java.version>
15         <spring-framework.version>4.3.0.RELEASE</spring-framework.version>
16         <cxfrtfrontend-jaxws.version>3.1.6</cxfrtfrontend-jaxws.version>
17
18     </properties>
19
20     <dependencyManagement>
21         <dependencies>
22             <dependency>
23                 <groupId>org.springframework</groupId>
24                 <artifactId>spring-framework-bom</artifactId>
25                 <version>${spring-framework.version}</version>
26                 <type>pom</type>
27                 <scope>import</scope>
28             </dependency>
29         </dependencies>
30     </dependencyManagement>
31
32     <dependencies>
33
34         <!-- Spring MVC -->
35         <dependency>
36             <groupId>org.springframework</groupId>
37             <artifactId>spring-webmvc</artifactId>
38         </dependency>
39
40         <dependency>
41             <groupId>org.apache.cxf</groupId>
42             <artifactId>cxfrtfrontend-jaxws</artifactId>
43             <version>${cxfrtfrontend-jaxws.version}</version>
44         </dependency>
45
46         <dependency>
47             <groupId>org.apache.cxf</groupId>
48             <artifactId>cxfrttransports-http</artifactId>
49             <version>${cxfrtfrontend-jaxws.version}</version>
50         </dependency>
51
52     </dependencies>
53
54     <build>
```



```
56     <plugins>
57     <plugin>
58         <artifactId>maven-compiler-plugin</artifactId>
59         <groupId>org.apache.maven.plugins</groupId>
60         <version>3.3</version>
61         <configuration>
62             <source>${java.version}</source>
63             <target>${java.version}</target>
64             <compilerArgument>-Xlint:all</compilerArgument>
65             <showWarnings>true</showWarnings>
66             <showDeprecation>true</showDeprecation>
67         </configuration>
68     </plugin>
69 </plugins>
70
71 </build>
72
73 </project>
```

4. web.xml

Библиотека Spring MVC используется исключительно для запуска приложения и удобного описания SOAP веб-сервиса. Использование Spring не является обязательным и приложение запускается с помощью небольшого метода с публикацией soap сервиса. Файл с настройками application-context.xml (с описанием SOAP веб-сервиса) добавлен в дескриптор развертывания. Так же был добавлен mvc-config.xml (не используется) в параметры инициализации init-param диспетчера сервлетов, чтобы приложение не ругалось на отсутствие файла с настройками (dispatcher-servlet.xml).

web.xml:



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5         http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
6     version="3.1">
7
8     <context-param>
9         <param-name>contextConfigLocation</param-name>
10        <param-value>/WEB-INF/config/applicationContext.xml</param-value>
11    </context-param>
12
13    <listener>
14        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
15    </listener>
16
17    <servlet>
18        <servlet-name>dispatcher</servlet-name>
19        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
20        <init-param>
21            <param-name>contextConfigLocation</param-name>
22            <param-value>/WEB-INF/config/mvc-config.xml</param-value>
23        </init-param>
24        <load-on-startup>1</load-on-startup>
25    </servlet>
26
27    <servlet-mapping>
28        <servlet-name>dispatcher</servlet-name>
29        <url-pattern>*.form</url-pattern>
30    </servlet-mapping>
31
32    <servlet>
33        <servlet-name>CXFServlet</servlet-name>
34        <servlet-class>org.apache.cxf.transport.servlet.CXFServlet</servlet-class>
35        <load-on-startup>2</load-on-startup>
36    </servlet>
37    <servlet-mapping>
38        <servlet-name>CXFServlet</servlet-name>
39        <url-pattern>/soap/*</url-pattern>
40    </servlet-mapping>
41
42
43 </web-app>
```

Для реализации веб-сервиса используется реализация JAX-WS API — Apache CXF. Создан сервлет и задан шаблон URL по которому он будет срабатывать (/soap/*).

5. application-context.xml

В файле конфигурации спринг описывается один soap веб сервис — его id, адрес и реализующий класс.



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:jaxws="http://cxf.apache.org/jaxws"
5       xsi:schemaLocation="
6         http://www.springframework.org/schema/beans
7         http://www.springframework.org/schema/beans/spring-beans.xsd
8         http://cxf.apache.org/jaxws
9         http://cxf.apache.org/schemas/jaxws.xsd">
10
11     <!--http://localhost:8080/soap/webserviceSEI?wsdl-->
12     <jaxws:endpoint id="webserviceSEI" address="/webserviceSEI"
13         implementor="ru.javastudy.ws.soap.HelloSoap"/>
14
15 </beans>
```

Здесь прописан веб сервис с id = helloRestService. Он будет доступен по ссылке, которая формируется из пути сервера + маппинг сервлета + адреса обработки сервиса (в данном случае совпадает с id).

6. SOAP сервис

Был создан простейший веб сервис по технологии SOAP.

Описание SOAP веб-сервиса находится в интерфейсе WebserviceSEI (service endpoint interface). Аннотация @WebMethod не является обязательной, но в ней можно прописывать дополнительные параметры и свойства для веб-сервиса.



```
1 package ru.javastudy.ws.soap;
2
3 import ru.javastudy.ws.model.Goods;
4
5 import javax.jws.WebMethod;
6 import javax.jws.WebParam;
7 import javax.jws.WebService;
8
9 /**
10  * Created for JavaStudy.ru on 11.06.2016.
11  */
12 /**
13  * The @WebService annotation on the implementation class lets CXF know which
14  * interface to use when creating WSDL. In this case its simply our HelloWorld interface.
15  */
16 @WebService
17 public interface WebserviceSEI {
18
19     @WebMethod//annotation optional and is mainly used to provide a name attribute to the public meth
20     od in wsdl
21     String testService();
22
23     @WebMethod
24     String sayHelloTo(@WebParam(name = "text") String text);
25
26     @WebMethod
27     Goods getGoods();
28 }
```

Реализация находится в классе HelloSoap:



```
1 package ru.javastudy.ws.soap;
2
3 import ru.javastudy.ws.model.Goods;
4 import javax.ws.WebService;
5
6 /**
7  * Created for JavaStudy.ru on 10.06.2016.
8  */
9 @WebService(endpointInterface = "ru.javastudy.ws.soap.WebserviceSEI",
10 serviceName = "HelloSoap")
11 public class HelloSoap implements WebserviceSEI {
12
13     @Override
14     public String testService() {
15         return "Hello from SOAP Webservice!";
16     }
17
18     @Override
19     public String sayHelloTo(String text) {
20         return "Hello to " + text;
21     }
22
23     @Override
24     public Goods getGoods() {
25         Goods goods = new Goods();
26         goods.setId(1);
27         goods.setName("Some goods test name");
28         return goods;
29     }
30 }
```

В аннотации указан интерфейс, которые описывает веб-сервис, а также имя этого сервиса. Реализованы тривиальные методы, которые позволят продемонстрировать работу soap сервиса.

7. Модель данных

Созданы два демонстрационных класса, которые будут преобразовываться в xml формат и передаваться клиенту. У классов задана аннотация `@XmlRootElement` с помощью которой присваивается имя для корневого xml элемента в ответе (см. далее запуск приложения).

Goods:





```
1 package ru.javastudy.ws.model;
2
3 import javax.xml.bind.annotation.XmlRootElement;
4 import java.io.Serializable;
5
6 /**
7  * Created for JavaStudy.ru on 11.06.2016.
8  */
9 @XmlRootElement(name = "goods")
10 public class Goods implements Serializable {
11
12     private int id;
13     private String name;
14
15     public Goods() {
16     }
17
18     public Goods(int id, String name) {
19         this.id = id;
20         this.name = name;
21     }
22
23     public int getId() {
24         return id;
25     }
26
27     public void setId(int id) {
28         this.id = id;
29     }
30
31     public String getName() {
32         return name;
33     }
34
35     public void setName(String name) {
36         this.name = name;
37     }
38
39     @Override
40     public String toString() {
41         return "Goods{" +
42             "id=" + id +
43             ", name=" + name + "\n" +
44             "}";
45     }
46 }
```

Document:



```
1 package ru.javastudy.ws.model;
2
3 import javax.xml.bind.annotation.XmlRootElement;
4 import java.util.List;
5
6 /**
7  * Created for JavaStudy.ru on 12.06.2016.
8  */
9 @XmlRootElement(name = "document")
10 public class Document {
11
12     private int id;
13     private String name;
14     private List<Goods> goodsList;
15
16     public Document() {
17     }
18
19     public Document(int id, String name, List<Goods> goodsList) {
20         this.id = id;
21         this.name = name;
22         this.goodsList = goodsList;
23     }
24
25     public int getId() {
26         return id;
27     }
28
29     public void setId(int id) {
30         this.id = id;
31     }
32
33     public String getName() {
34         return name;
35     }
36
37     public void setName(String name) {
38         this.name = name;
39     }
40
41     public List<Goods> getGoodsList() {
42         return goodsList;
43     }
44
45     public void setGoodsList(List<Goods> goodsList) {
46         this.goodsList = goodsList;
47     }
48 }
```

8. SOAP клиент

Для тестирования нашего сервиса был создан код, имитирующий клиентскую сторону 

JavaStudyWS:

```
1 package ru.javastudy.ws.main;
2
3 import org.apache.cxf.jaxws.JaxWsProxyFactoryBean;
4 import ru.javastudy.ws.model.Goods;
5 import ru.javastudy.ws.soap.WebserviceSEI;
6
7 /**
8  * Created for JavaStudy.ru on 09.06.2016.
9  */
10 public class JavaStudyWS {
11
12     public static void main(String[] args) {
13         testSOAPFromClient();
14     }
15
16     /**
17     * create client and test soap service
18     */
19     private static void testSOAPFromClient() {
20         String soapServiceUrl = "http://localhost:8080/soap/webServiceSEI";
21
22         JaxWsProxyFactoryBean factoryBean = new JaxWsProxyFactoryBean();
23         factoryBean.setServiceClass(WebserviceSEI.class);
24         factoryBean.setAddress(soapServiceUrl);
25
26         WebserviceSEI webserviceSEI = (WebserviceSEI) factoryBean.create();
27
28         Goods result = webserviceSEI.getGoods();
29         System.out.println("Result: " + result);
30     }
31
32 }
```

С помощью фабрики `JaxWsProxyFactoryBean` создается soap веб-сервис доступные по адресу `soapServiceURL`. Далее вызывается один из методов, описанных в интерфейсе сервиса.

9. Запуск веб сервиса

После запуска веб сервиса его описание можно посмотреть по ссылке `http://localhost:8080/soap/webServiceSEI?wsdl`.





(<http://javastudy.ru/wp-content/uploads/2016/06>

/WebServiceWSDL.png)

Если запустить main метод класса JavaStudyWS, то мы увидим ответ веб-сервиса в консоли.

1 | Result: Goods{id=1, name='Some goods test name'}



(<http://javastudy.ru/wp-content/uploads/2016/06>

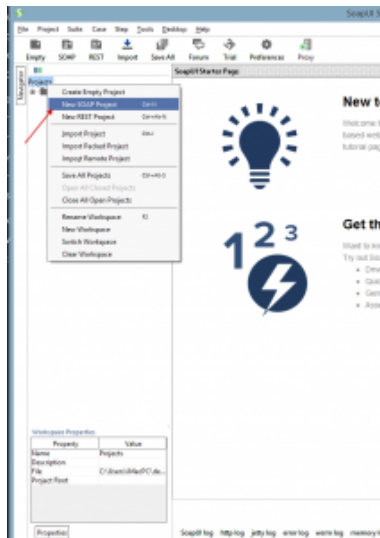
/soapRun.png)

9.1 SoapUI

SOAP веб-сервисы удобно тестировать с помощью утилиты SoapUI.

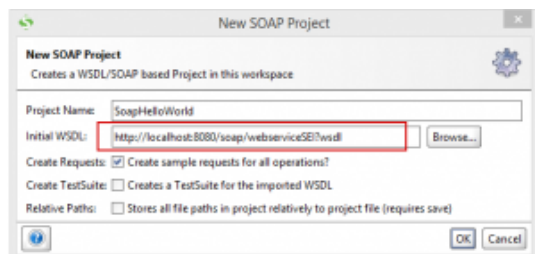
Добавляем новый SOAP проект.





(<http://javastudy.ru/wp-content/uploads/2016/06/SoapUI1.png>)

Указываем путь к wsdl описанию нашего сервиса.



(<http://javastudy.ru/wp-content/uploads/2016/06>

/SoapUI2.png)

Запускаем метод веб-сервиса.



(<http://javastudy.ru/wp-content/uploads/2016/06>

/SoapUI3.png)

Таким образом можно создать простейший SOAP веб сервис java.

Исходный код

SOAP HelloWorld (<http://javastudy.ru/wp-content/uploads/2016/06/SOAP-HelloWorld.rar>)

Share (<http://www.hupso.com/share/>)





([http://www.hupso.com/share/add.php?service=twitter&title=SOAP%20java%20%D0%B2%D0%B5%D0%B1%20%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81.%20%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80%20Hello%20World%20example&url=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F&](http://www.hupso.com/share/add.php?service=twitter&title=SOAP%20java%20%D0%B2%D0%B5%D0%B1%20%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81.%20%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80%20Hello%20World%20example&url=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F&via=java2study)

via=java2study) ([http://www.hupso.com/share/add.php?service=facebook&title=SOAP%20java%20%D0%B2%D0%B5%D0%B1%20%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81.%20%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80%20Hello%20World%20example&url=http%3A%2F%2Fjavastudy.ru%2Fweb-](http://www.hupso.com/share/add.php?service=facebook&title=SOAP%20java%20%D0%B2%D0%B5%D0%B1%20%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81.%20%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80%20Hello%20World%20example&url=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F)

services%2Fsoap-java-hello-world-example%2F) (<http://www.hupso.com/share/add.php?service=vkontakte&title=SOAP%20java%20%D0%B2%D0%B5%D0%B1%20%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81.%20%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80%20Hello%20World%20example&url=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F>)

8

Views: 2 939

Рубрики [Web services \(http://javastudy.ru/category/web-services/\)](http://javastudy.ru/category/web-services/)

Комментарии (5) для “SOAP java веб сервис. Пример Hello World example”

*sir.mllmn:*

19.09.2016 в 21:06 (<http://javastudy.ru/web-services/soap-java-hello-world-example/#comment-482>)

«Для реализации веб-сервиса используется реализация Jax-RS API — Apache CXF. Создан сервлет и задан шаблон URL по которому он будет срабатывать (/rest/*).» — наверное, ошибка копи паста из примера рестфул вс.

Войдите, чтобы ответить (http://javastudy.ru/wp-login.php?redirect_to=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F)

*Nick V. (<http://javastudy.ru/>):*

20.09.2016 в 22:36 (<http://javastudy.ru/web-services/soap-java-hello-world-example/#comment-488>)

именно так)

Войдите, чтобы ответить (http://javastudy.ru/wp-login.php?redirect_to=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F)





Евгения:

23.12.2016 в 23:50 (<http://javastudy.ru/web-services/soap-java-hello-world-example/#comment-1300>)

Не работает. Запускаю, а мне в ответ

Caused by: java.net.ConnectException: ConnectException invoking http://localhost:8080/soap/webserviceSEI (http://localhost:8080/soap/webserviceSEI): Connection refused: connect at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)

Войдите, чтобы ответить (http://javastudy.ru/wp-login.php?redirect_to=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F)



Nick V. (<http://javastudy.ru>):

24.12.2016 в 22:46 (<http://javastudy.ru/web-services/soap-java-hello-world-example/#comment-1312>)

Из всего стектрейса ясно что проблема с

Caused by: java.net.ConnectException: ConnectException invoking http://localhost:8080/soap/webserviceSEI (http://localhost:8080/soap/webserviceSEI): Connection refused: connect

Войдите, чтобы ответить (http://javastudy.ru/wp-login.php?redirect_to=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F)



Надежда:

18.05.2017 в 22:41 (<http://javastudy.ru/web-services/soap-java-hello-world-example/#comment-3284>)

А что в файле mvс—config.xml? без него не запускается

18-May-2017 22:16:06.335 SEVERE [localhost-startStop-1]
org.springframework.web.context.ContextLoader.initWebApplicationContext Context initialization failed
org.springframework.beans.factory.BeanDefinitionStoreException: IOException parsing XML document from ServletContext resource [/WEB-INF/config/applicationContext.xml]; nested exception is java.io.FileNotFoundException: Could not open ServletContext resource [/WEB-INF/config/applicationContext.xml]

Войдите, чтобы ответить (http://javastudy.ru/wp-login.php?redirect_to=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F)



Добавить комментарий

Для отправки комментария вам необходимо авторизоваться (http://javastudy.ru/wp-login.php?redirect_to=http%3A%2F%2Fjavastudy.ru%2Fweb-services%2Fsoap-java-hello-world-example%2F).

💡 RESTFul java веб сервис. Пример Hello World example (<http://javastudy.ru/web-services/restful-java-hello-world-example/>)

💡 HandlerChain в java SOAP веб-сервисе. Применение цепочки handler'ов (<http://javastudy.ru/web-services/handlerchain-java-soap/>)

Поиск

Подписывайтесь



(<https://twitter.com/java2study>)



(<https://www.facebook.com/java2study>)



(<https://vk.com/java2study>)

Разделы

AngularJS (11) (<http://javastudy.ru/category/angularjs/>)

database-tools (2) (<http://javastudy.ru/category/database-tools/>)

GWT (8) (<http://javastudy.ru/category/gwt/>)

Hibernate (11) (<http://javastudy.ru/category/hibernate/>)

Interview (27) (<http://javastudy.ru/category/interview/>)

java (1) (<http://javastudy.ru/category/java/>)

JSF 2 (4) (<http://javastudy.ru/category/jsf-2/>)

PrimeFaces (1) (<http://javastudy.ru/category/jsf-2/primefaces/>)



jUnit (4) (<http://javastudy.ru/category/junit/>)

Log4j (1) (<http://javastudy.ru/category/log4j/>)

Patterns (1) (<http://javastudy.ru/category/patterns/>)

Spring Data JPA (16) (<http://javastudy.ru/category/spring-data-jpa/>)

Spring MVC (36) (<http://javastudy.ru/category/spring-mvc/>)

Spring Security (5) (<http://javastudy.ru/category/spring-security/>)

Spring Web Flow (20) (<http://javastudy.ru/category/spring-web-flow/>)

tools (1) (<http://javastudy.ru/category/tools/>)

Tutorials (1) (<http://javastudy.ru/category/tutorials/>)

Web services (9) (<http://javastudy.ru/category/web-services/>)

webapp (3) (<http://javastudy.ru/category/webapp/>)

Разное (1) (<http://javastudy.ru/category/default/>)

Нашли ошибку?

ихний магазин

Выделите и нажмите CTRL+ENTER :)

Недавние записи

Databene Benerator — автоматическая генерация тестовых данных (<http://javastudy.ru/database-tools/databene-benerator-generate-test-data/>)

📅 04.02.2017

Подключение к базе данных Oracle из IntelliJ IDEA (<http://javastudy.ru/webapp/oracle-connect-intellij-idea/>)

📅 04.02.2017

Flyway — подключение, настройка библиотеки и плагина flyway-maven-plugin. (<http://javastudy.ru/database-tools/flyway-start-maven-plugin/>)

📅 04.02.2017



JUnit Listener — добавление слушателей в юнит-тесты (<http://javastudy.ru/junit/junit-listener/>)

📅 02.07.2016

Параметризованные тесты в JUnit. Использование @Parameterized.Parameters (<http://javastudy.ru/junit/parameterized-test-junit/>)

📅 02.07.2016

JUnit — Suite тест. Запуск нескольких тестов с помощью аннотаций @Suite и @RunWith (<http://javastudy.ru/junit/junit-suite-tests/>)

📅 02.07.2016

JUnit — введение в юнит-тесты. Пример JUnit Hello world (<http://javastudy.ru/junit/junit-hello-world/>)

📅 02.07.2016

Включение адресации WS-Addressing в SOAP java веб-сервисе (<http://javastudy.ru/web-services/ws-addressing-soap/>)

📅 02.07.2016

Валидация данных веб-сервиса на стороне сервера с помощью reflection (<http://javastudy.ru/web-services/validation-with-reflection/>)

📅 02.07.2016

Обработка ошибок в SOAP веб-сервисе на стороне клиента (<http://javastudy.ru/web-services/exceptions-soap-clientside/>)

📅 23.06.2016

[JavaStudy.ru \(http://javastudy.ru/main-page/\)](http://javastudy.ru/main-page/)

[About \(http://javastudy.ru/about/\)](http://javastudy.ru/about/)

[Advertising \(http://javastudy.ru/advertising/\)](http://javastudy.ru/advertising/)

[Contacts \(http://javastudy.ru/contact-us/\)](http://javastudy.ru/contact-us/)

[Sitemap \(http://javastudy.ru/sitemap/\)](http://javastudy.ru/sitemap/)

[Frameworks \(http://javastudy.ru/frameworks/\)](http://javastudy.ru/frameworks/)

[GWT \(http://javastudy.ru/frameworks/gwt/\)](http://javastudy.ru/frameworks/gwt/)



Hibernate (<http://javastudy.ru/frameworks/hibernate/>)

JSF 2.0 (<http://javastudy.ru/frameworks/jsf-2-0/>)

Spring (<http://javastudy.ru/frameworks/spring/>)

Unit Test (<http://javastudy.ru/frameworks/unit-test/>)

Reference (<http://javastudy.ru/reference/>)

Interview (<http://javastudy.ru/reference/interview/>)

Books (<http://javastudy.ru/reference/books/>)

Tools (<http://javastudy.ru/tools/>)

Database tools (<http://javastudy.ru/tools/database-tools/>)

Apache Maven (<http://javastudy.ru/tools/apache-maven/>)

Log4j 2 (<http://javastudy.ru/tools/log4j/>)

jQuery (<http://javastudy.ru/tools/jquery/>)

Javastudy.ru © 2017 - При любом использовании материала, пожалуйста, указывайте ссылку на данный ресурс:)

