

Vol II Issue VIII Feb 2013

Impact Factor : 1. 2018

ISSN No :2231-5063

# Monthly Multidisciplinary Research Journal

## *Golden Research Thoughts*

Chief Editor  
Dr.Tukaram Narayan Shinde

Publisher  
Mrs.Laxmi Ashok Yakkaldevi

Associate Editor  
Dr.Rajani Dalvi

Honorary  
Mr.Ashok Yakkaldevi

## IMPACT FACTOR : 1. 2018

### Welcome to ISRJ

**RNI MAHMUL/2011/38595**

**ISSN No.2230-7850**

Indian Streams Research Journal is a multidisciplinary research journal, published monthly in English, Hindi & Marathi Language. All research papers submitted to the journal will be double - blind peer reviewed referred by members of the editorial Board readers will include investigator in universities, research institutes government and industry with research interest in the general subjects.

### *International Advisory Board*

Flávio de São Pedro Filho Federal University of Rondonia, Brazil	Mohammad Hailat Dept. of Mathematical Sciences, University of South Carolina Aiken, Aiken SC 29801	Hasan Baktir English Language and Literature Department, Kayseri
Kamani Perera Regional Centre For Strategic Studies, Sri Lanka	Abdullah Sabbagh Engineering Studies, Sydney	Ghayoor Abbas Chotana Department of Chemistry, Lahore University of Management Sciences [ PK ]
Janaki Sinnasamy Librarian, University of Malaya [ Malaysia ]	Catalina Neculai University of Coventry, UK	Anna Maria Constantinovici AL. I. Cuza University, Romania
Romona Mihaila Spiru Haret University, Romania	Ecaterina Patrascu Spiru Haret University, Bucharest	Horia Patrascu Spiru Haret University, Bucharest, Romania
Delia Serbescu Spiru Haret University, Bucharest, Romania	Loredana Bosca Spiru Haret University, Romania	Ilie Pinteau, Spiru Haret University, Romania
Anurag Misra DBS College, Kanpur	Fabricio Moraes de Almeida Federal University of Rondonia, Brazil	Xiaohua Yang PhD, USA
Titus Pop	George - Calin SERITAN Postdoctoral Researcher	Nawab Ali Khan College of Business Administration

### *Editorial Board*

Pratap Vyamktrao Naikwade ASP College Devrukh,Ratnagiri,MS India	Iresh Swami Ex - VC. Solapur University, Solapur	Rajendra Shendge Director, B.C.U.D. Solapur University, Solapur
R. R. Patil Head Geology Department Solapur University, Solapur	N.S. Dhaygude Ex. Prin. Dayanand College, Solapur	R. R. Yaliker Director Managment Institute, Solapur
Rama Bhosale Prin. and Jt. Director Higher Education, Panvel	Narendra Kadu Jt. Director Higher Education, Pune	Umesh Rajderkar Head Humanities & Social Science YCMOU, Nashik
Salve R. N. Department of Sociology, Shivaji University, Kolhapur	K. M. Bhandarkar Praful Patel College of Education, Gondia	S. R. Pandya Head Education Dept. Mumbai University, Mumbai
Govind P. Shinde Bharati Vidyapeeth School of Distance Education Center, Navi Mumbai	Sonal Singh Vikram University, Ujjain	Alka Darshan Shrivastava Shaskiya Snatkottar Mahavidyalaya, Dhar
Chakane Sanjay Dnyaneshwar Arts, Science & Commerce College, Indapur, Pune	G. P. Patankar S. D. M. Degree College, Honavar, Karnataka	Rahul Shriram Sudke Devi Ahilya Vishwavidyalaya, Indore
Awadhesh Kumar Shirotriya Secretary, Play India Play (Trust),Meerut	Maj. S. Bakhtiar Choudhary Director,Hyderabad AP India.	S.KANNAN Ph.D , Annamalai University,TN
	S.Parvathi Devi Ph.D.-University of Allahabad	Satish Kumar Kalhotra
	Sonal Singh	

**Address:-Ashok Yakkaldevi 258/34, Raviwar Peth, Solapur - 413 005 Maharashtra, India  
Cell : 9595 359 435, Ph No: 02172372010 Email: ayisrj@yahoo.in Website: www.isrj.net**



## TESTING WEB SERVICES – A CLIENT SIDE PERSPECTIVE

PRASANTH YALLA AND L S S REDDY

Associate Professor , K.L.University, Guntur dist, AP India.  
Director,LBR College of EngineeringMylavaram, Krishna District. AP, India.

### Abstract:

*At times we don't have a specific testing strategy to consider the idea of testing web services but now a days so many tools which are available on-line. These tools may or may not concentrate on functional testing because there is no rule that we need to consider only functional testing; as it is only a part of the testing phenomenon. There is a tool called SoapUI which allows the user to work with the web services and to execute the test cases in a different environment. We need to believe that a web service does not have GUI and they use XML encoded data for communication. From client side testing it is very important because the test cases generated should be definitely in XML format. In this we define how to perform web service testing from client side, client file, implementation file, proxy class and a service locator file which talks about their respective tasks. We can describe these services by using a language called WSDL (Web Services Description Language). Moreover client side scripting is a collection of programs that are usually run at client side rather at server-side i.e web server. Client side scripts are java scripts because execution is done by the browser (client). Server side scripts are CGI scripts written in PHP, perl, ASP.net etc*

### KEY-WORDS:

SoapUI, XML, WSDL.

### 1.INTRODUCTION

Testing is considered to be the most crucial part of the software development life cycle. It is very important because it really helps the users to measure different parameters of quality like correctness and robustness of a particular web service. Manual testing is so tiresome that we have to spent lot of time on analysis, identification and testing of test cases; where some tools are needed to automate the generation of test cases and implementation for web services. There are so many tools available in the market to generate test cases automatically and also which helps the user in analyzing the combination or request and response. Automated testing always concentrates on perfect generation of test cases automatically and executes such generated test cases. This kind of testing will save time and costs less effort. Sometimes we can consider this testing as model-based where initially the basic functional requirements are designed and then based on the designed and well-framed model automatically the test cases will be generated.

When it comes to the manual testing, the results get compared to the expected behavior and all such observations will be recorded. Whereas in automated testing, all the cases are pre-defined. Once comparison is over, it becomes very easy to extend its functionality which is not possible in manual testing.

## 2. HOW TO PERFORM WEB SERVICES TESTING

Testing web services from client side involves:

1. Creating a java project.
2. Consideration of web services.
3. Selecting Web service client.
4. Creating Web service project.
5. Creation of Classes.

The information related to the client will be considered as input to the web service. The Figure 2.1 a: shown is the interface which acts as barrier between the consumer and producer. When we think about component level testing; black box and white box testing comes into the picture which is necessary for each function. The practical test cases which are designed basically depend on boundary value analysis, path testing, forced error testing and equivalence partitioning. We can create a web service client by eclipse going to menu File click New click Other click Web Services click Web Service Client.

The following is the interface generated by the Web service Client.

```

/**
 * Calculator.java
 */
 * This file was auto-generated from WSDL
 * by the Apache Axis 1.4 Apr 22, 2006 (06:55:48 PDT) WSDL2Java emitter.
 */
package com.prasanth.utilities;

public interface Calculator extends java.rmi.Remote {
    public java.lang.String additionOperation(java.lang.String in1, java.lang.String in2) throws java.rmi.RemoteException;
    public java.lang.String subtractionOperation(java.lang.String in1, java.lang.String in2) throws java.rmi.RemoteException;
    public java.lang.String multiplicationOperation(java.lang.String in1, java.lang.String in2) throws java.rmi.RemoteException;
    public java.lang.String divisionOperation(java.lang.String in1, java.lang.String in2) throws java.rmi.RemoteException;
}

```

**Figure 2.1 a: Calculator.java - interface**

Client side testing is done by choosing the conceptual web services and the web service client. The java file called calculator.java is a client which describes the functionality of the web service which is acting as client. In order to create a client from the WSDL, you have to create both Configuration file and Java files both from WSDL. Configuration of the client talks about the bindings which allow the consumer or user to communicate with the host system.

The Figure 2.1 a: talks about the Calculator acting as interface where java.lang.String can also be defined as class. We know that Strings are objects and particularly they are the instances of the class java.lang.String. Strings start their count from 0 (Zero). And its implementation is made known in Figure 2.1 b.

```

package com.prasanth.utilities;
/**
 * @author PRASANTH
 */
public class Calculator {
    private static String response;
    /**
     * @param in1
     * @param in2
     * @return
     */
    public static String additionOperation(String in1, String in2) {
        response = (Integer.parseInt(in1)+Integer.parseInt(in2))+"";
        return response;
    }
    /**
     * @param in1
     * @param in2
     * @return
     */
    public static String subtractionOperation(String in1, String in2) {
        response = (Integer.parseInt(in1)-Integer.parseInt(in2))+"";
        return response;
    }
    /**
     * @param in1
     * @param in2
     * @return
     */
    public static String multiplicationOperation(String in1, String in2) {
        response = (Integer.parseInt(in1)*Integer.parseInt(in2))+"";
        return response;
    }
    /**
     * @param in1
     * @param in2
     * @return
     */
    public static String divisionOperation(String in1, String in2) {
        response = (Integer.parseInt(in1)/Integer.parseInt(in2))+"";
        return response;
    }
}

```

Figure 2.1 b: Calculator.java - Implementation

```

package com.prasanth.utilities;
public class CalculatorProxy implements com.prasanth.utilities.Calculator {
    private String _endpoint = null;
    private com.prasanth.utilities.Calculator calculator = null;
    public CalculatorProxy() {
        _initCalculatorProxy();
    }
    public CalculatorProxy(String endpoint) {
        _endpoint = endpoint;
        _initCalculatorProxy();
    }
    /**
     *
     */
    private void _initCalculatorProxy() {
        try {
            calculator = (new com.prasanth.utilities.CalculatorServiceLocator()).getCalculator();
            if (calculator != null) {
                if (_endpoint != null)
                    ((javax.xml.rpc.Stub)calculator)._setProperty("javax.xml.rpc.service.endpoint.address", _endpoint);
                else
                    _endpoint = (String)((javax.xml.rpc.Stub)calculator)._getProperty("javax.xml.rpc.service.endpoint.address");
            }
        }
    }
}

```

Figure 2.2: CalculatorProxy.java

Figure 2.2 shows the intervention of CalculatorProxy.java defined as a proxy class which usually implements different interfaces only at run time. Proxy class is often defined as public. It implements only those interfaces which are mentioned at the time of creation. The javax.xml.rpc.Stub is the interface for all the classes of type Stub. There are different interfaces like Stub, Call, and Service which is also a class. This can also be considered a package which commonly contains core JAX-RPC APIs for the client.

```

/* CalculatorServiceLocator.java
*/
package com.prasanth.utilities;
/**
 * @author PRASANTH
 */
public class CalculatorServiceLocator extends org.apache.axis.client.Service implements com.prasanth.utilities.CalculatorService {
    public CalculatorServiceLocator() {
    }
    public CalculatorServiceLocator(org.apache.axis.EngineConfiguration config) {
        super(config);
    }
    public CalculatorServiceLocator(java.lang.String wsdlLoc, javax.xml.namespace.QName sName) throws javax.xml.rpc.ServiceException {
        super(wsdlLoc, sName);
    }
    // Use to get a proxy class for Calculator
    private java.lang.String Calculator_address = "http://localhost:9090/WebServiceProject/services/Calculator";
    public java.lang.String getCalculatorAddress() {
        return Calculator_address;
    }
    // The WSDL service name defaults to the port name.
    private java.lang.String CalculatorWSDServiceName = "Calculator";

```

**Figure 2.3: CalculatorServiceLocator**

The ServiceLocator is a pattern used to identify the Enterprise Java Beans often called EJB and destinations of Java Message Service-JMS which are often called service components. The clients of J2EE communicate with them; in the process of communication they have to create their own components. Sometimes, these clients are capable of locating their service components. In order to down the complexity of the code, several clients can reuse the service locator. This behavior is shown in Figure 2.3



```

* CalculatorSoapBindingStub.java
package com.prasanth.utilities;

public class CalculatorSoapBindingStub extends org.apache.axis.client.Stub implements com.prasanth.utilities.Calculator {
    private java.util.Vector cachedSerClasses = new java.util.Vector();
    private java.util.Vector cachedSerNames = new java.util.Vector();
    private java.util.Vector cachedSerFactories = new java.util.Vector();
    private java.util.Vector cachedDeserFactories = new java.util.Vector();

    static org.apache.axis.description.OperationDesc [] _operations;

    static {
        _operations = new org.apache.axis.description.OperationDesc[4];
        _initOperationDesc();
    }

    private static void _initOperationDesc(){
        org.apache.axis.description.OperationDesc oper;
        org.apache.axis.description.ParameterDesc param;
        oper = new org.apache.axis.description.OperationDesc();
        oper.setName("additionOperation");
        param = new org.apache.axis.description.ParameterDesc(new javax.xml.namespace.QName("http://utilities.prasanth.com", "in1"), org.apache.axis.description.ParameterDesc.MULTI, false, null, null, null, null);
        oper.addParameter(param);
        param = new org.apache.axis.description.ParameterDesc(new javax.xml.namespace.QName("http://utilities.prasanth.com", "in2"), org.apache.axis.description.ParameterDesc.MULTI, false, null, null, null, null);
        oper.addParameter(param);
        oper.setReturnType(new javax.xml.namespace.QName("http://www.w3.org/2001/XMLSchema", "string"));
        oper.setReturnClass(java.lang.String.class);
        oper.setReturnQName(new javax.xml.namespace.QName("http://utilities.prasanth.com", "additionOperationReturn"));
        oper.setStyle(org.apache.axis.constants.Style.WRAPPED);
        oper.setUse(org.apache.axis.constants.Use.LITERAL);
        _operations[0] = oper;
    }
}

```

Figure 2.4: CalculatorSoapBindingStub.java

More often SoapBindingStubs are automatically generated from the WSDL. We have to remember that all soap tool kits cannot contain these classes. These are client side stubs. For every specification in WSDL generating a SoapBindingStub.java is done by WSDL2Java which is shown in Figure 2.4. It is commonly used like a stub at the client side. In order to generate the stubs you can use ApacheAxis and WSDL2Java.

```

package com.prasanth.utilities;
import java.rmi.RemoteException;

public class Testing {
    public static void main(String[] args) throws RemoteException {
        CalculatorProxy proxy = new CalculatorProxy();
        proxy.setEndpoint("http://localhost:9090/WebserviceProject/services/Calculator");
        String resp = proxy.additionOperation("10", "20");
        System.out.println(resp);
    }
}

```

Figure 2.5: Testing.java

Here, in the Figure 2.5 the class defined is testing. The proxy is new CalculatorProxy(). To communicate with the APIs of web services, a WS Proxy client will be used by JWS client. Invocation of WS APIs as local method is the general attitude of a proxy client. WS interfaces are always described by WSDL files. SetEndpoint() is a method in the proxy class, by using this we can change the url for a particular web service. Normally, the endpoint url is defined towards local host. If you won't set the endpoint url, it will consider the url in wsdl.

### 3. CONCLUSION

No matter what ever may be the testing strategy used the testing of web services made revolutionary changes. In this context so many testing approaches can be followed and we've concentrated only on functional testing and how this could be done. Moreover on the client side perspective testing is so interesting about the clients of J2EE communicates with service components; in the process of communication they have to create their own components. Sometimes, these clients are capable of locating their service components. Sometimes we need to create classes to enhance the testing stability of web services. The context shown above is only a viewpoint from client-side.

### REFERENCES:

- [1] Antonia Bertolino. "Software Testing Research: Achievements, Challenges, Dreams." Future of Software Engineering (FOSE'07), IEEE Computer Society. (2007).
- [2] White Paper on "Web Services", Sonera Plaza| Media Lab (2002)
- [3] Waseem Roshen "Enterprise Service Bus with USB-Like Universal Ports", in the proceedings of Ninth IEEE European Conference on Web Services- IEEE Computer Society, pages 177-183 (2011).
- [4] Mike P. Papazoglou · Willem-Jan van den Heuvel "Service oriented architectures: approaches, technologies and research issues" The VLDB Journal- Springer, Pages 389-415, (2007).
- [5] <http://www.applabs.com/html/soatestingchallenges.html>.
- [6] Gerardo Canfora and Massimiliano Di Penta "SOA: Testing and Self-Checking", Proc. of International Workshop on Web Services Modeling and Testing (WS-MaTe 2006), Pages 1-12 (2006).
- [7] <http://www.c-sharpcorner.com/UploadFile/5a0e30/functional-testing-part-1/>
- [8] Kamalsinh F Chavda "Anatomy of a Web Service", In the Proc of "CCSC: Southeastern Conference", pages 124-134 (January 2004).
- [9] White Paper By Kevin J. Ma On 'Web Services: What's Real and What's Not?', IT Pro, IEEE Computer Society, Pages 14-21 (March/April 2005).
- [10] Shrideep Pallickara, Geoffrey Fox and Lee Pallickara, "An Analysis of Reliable Delivery Specifications for Web Services", In the Proc of "International Conference on Information Technology: Coding and Computing (ITCC'05)", IEEE Computer Society (2005).
- [11] <http://www.ibm.com/developerworks/xml/library/x-namcar/index.html>
- [12] Thomas Y. Lee, David W. Cheung "XML Schema Computations: Schema Compatibility Testing and Sub schema Extraction", CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada. Pages 839-848 (October 2010).
- [13] G.M Tere, B.T Jadhav, "Designing Efficient XML Web Services" In the Proc of "International Conference and Workshop on Emerging Trends in Technology (ICWET 2011) – TCET, Mumbai, India", pages 193-197, February 25–26, (2011).
- [14] Chunyan Ma<sup>1,2</sup>, Chenglie Du<sup>2</sup>, Tao Zhang<sup>1</sup>, Fei Hu, Xiaobin Cai. "WSDL-Based Automated Test Data Generation for Web Service." 2008 International Conference on Computer Science and Software Engineering.
- [15] D. Cohen, S.R. Dalal, M.L. Fredman, and G.C. Patton, "The AETG system: an approach to testing based on combinatorial Design", IEEE Transactions On Software Engineering, 23(7), July 1997.
- [16] Massimiliano Di Penta etc, Test and Analysis of Web Services, Publisher: Springer Berlin Heidelberg, (2007).
- [17] Evan Martin, Suranjana Basu, Tao Xie "WebSob: A Tool for Robustness Testing of Web Services", In the Proc of 29th International Conference on Software Engineering (ICSE'07 Companion), pages 1-2 (2007).
- [18] Wuzhi Xu, Jeff Offutt and Juan Luo "Testing Web Services by XML Perturbation" supported by "U.S. National Air and Space Administration, GSFC" pages 1-8
- [19] Avik Sinha, Amit Paradkar, "Model Based Functional Conformance Testing of Web Services Operating on Persistent Data", TAVWEB, Pages 1-6 (July 17, 2006).
- [20] White Paper: By Erin Cavanaugh on "Web services: Benefits, challenges, and a unique, visual



development solution”, Altova, Pages 1-21 (2009).



**PRASANTH YALLA** received his B.Tech. Degree from Acharya Nagarjuna University, Guntur (Dist), India in 2001, the M.Tech. degree in Computer Science and Engineering from Acharya Nagarjuna University in 2004, and presently pursuing the Ph.D. in Web services and SOA from Acharya Nagarjuna University, Guntur (Dist), India. He was an associate professor, with Department of Information Science and Technology in KL University, from 2004 to 2010. Presently he is working as associate professor, with the department of Freshman Engineering from 2011 in KL University. His research interests include Testing tools, Software Engineering, Web services and SOA. He taught several subjects like Multimedia technologies, Distributed Systems, Advanced Software Engineering, Object Oriented Analysis and design, C programming, Operating Systems, Database management systems, UML etc. He is the Life member of CSI



**L.S.S. REDDY** received his Ph.D from BITS, PILANI and presently Working as DIRECTOR, Lakireddy Bali Reddy College of Engineering, Mylavaram Krishna district. He is having more than 30 years of experience in teaching. Previously he Worked in various prestigious institutions like CBIT, Birla Institute of Technology & Science (BITS), PILANI, MVSU (Matrusri) Engineering College, Hyderabad, V.R. Siddhartha Engineering College, Vijayawada, as a principal, K.L. College of Engineering. He got many AWARDS/HONORS like Chairman Academic Council KLCE (Autonomous), Recognized guide for guiding scholars of Ph.D in Computer Science of Mysore University, Acharya Nagarjuna University & JNTU, Hyderabad, Chairman, board of studies Computer Science and Engineering, and Information Technology, Acharya Nagarjuna University, Chairman, CSI Chapter- Vijayawada. Senior Member, IEEE Computer Society. Member, Expert Panel of DOEACC Society and many more.

# Publish Research Article International Level Multidisciplinary Research Journal For All Subjects

Dear Sir/Mam,

We invite unpublished research paper.Summary of Research Project,Theses,Books and Books Review of publication,you will be pleased to know that our journals are

## Associated and Indexed,India

- \* International Scientific Journal Consortium Scientific
- \* OPEN J-GATE

## Associated and Indexed,USA

- EBSCO
- Index Copernicus
- Publication Index
- Academic Journal Database
- Contemporary Research Index
- Academic Paper Databse
- Digital Journals Database
- Current Index to Scholarly Journals
- Elite Scientific Journal Archive
- Directory Of Academic Resources
- Scholar Journal Index
- Recent Science Index
- Scientific Resources Database

Golden Research Thoughts  
258/34 Raviwar Peth Solapur-413005,Maharashtra  
Contact-9595359435  
E-Mail-ayisrj@yahoo.in/ayisrj2011@gmail.com  
Website : www.isrj.net