



Science

## **REGISTRY FOR RESTful WEB SERVICE: RESTRegistry**

**Dr. Digvijaysinh Rathod** \*<sup>1</sup>

\*<sup>1</sup> Institute of Forensic Science, Gujarat Forensic Sciences University, Gandhiangar, Gujarat, India

---

### **Abstract**

SOAP/WSDL and REST are two architectural styles to build web services. SOAP/WSDL follows operation centric and REST follows resource centric approach. SOAP/WSDL has standard registry where producer can register service specification and consumer can discover service endpoint using UDDI. Recently RESTful web service gained momentum in development of distributed application and research community already started discussion about RESTful based Service Oriented Architecture(SOA) but there is no standard central repository exist where producer can register and consumer can search RESTful service based on Quality of Services or based on their specification. In this research paper, we proposed framework: XML based RESTRegistry – central repository with Quality of Service through which producer can easily register services and consumer can search it easily. We also developed Travel Schedule benchmark application to test proposed framework in real world.

**Keywords:** RESTRegistry; Central Registry; REST; RESTful; SOAP; Web Service; Composition; WSDL; HTTP.

**Cite This Article:** Dr. Digvijaysinh Rathod. (2017). “REGISTRY FOR RESTful WEB SERVICE: RESTRegistry.” *International Journal of Research - Granthaalayah*, 5(7), 128-135. <https://doi.org/10.5281/zenodo.835523>.

---

### **1. Introduction**

RESTful HTTP-based and SOAP/WSDL-based services are two architectural styles for building distributed system and web services can be described as software entities that are capable of delivering certain functionalities over a network as per WWW consortium. Representational State Transfer (REST) is an architectural style that first defined by Roy Fielding [1] for distributed system such as World Wide Web. This architectural style describes six constraints: Client-server, Stateless, Cacheable, Layered system, Code on demand (optional) and Uniform interface. Resources and representations is key concepts of RESTful web services where resources are abstraction of information and each of these resources are named with unique identifiers like URIs. When browser sends HTTP GET request to web application for targeted resources [13], it receives response in form of HTML representation of the resource's state. RESTful web services uses HTTP methods i.e., GET, PUT, POST and DELETE for CRUD [2,7]

(Create, Read, Update and Delete) to access and manipulate resources. RESTful web services are light weight, easy to deploy, inexpensive to host and easy to maintain as compare to SOAP/WSDL web services. Because these feature it is suitable for rapid development of thin client application, mobile application and web application and encourages architects or research communities to apply REST concepts to SOA.

Service-oriented architecture (SOA) and development is a paradigm where software components are created with concise interfaces, and each component performs a discrete set of related functions. With its well-defined interface and contract for usage, each component, provides a service to other software components. This is analogous to an accountant who provides a service to a business, even though that service consists of many related functions bookkeeping, tax filing, investment management, and so on.

There are no technology requirements or restrictions with SOA. We can build a service in any language with standards such as CORBA, remote procedure calls (RPC), or XML. Although SOA has been around as a concept for years [12, 14], its vague definition makes it difficult to identify. The client/server development model of the early '90s was a simple example of an SOA-based approach to software development.

A web service is an example of an SOA with a well-defined set of implementation choices. In general, the technology choices are SOAP and the Web Service Definition Language (WSDL); both XML-based. WSDL describes the interface (the "contract"), while SOAP describes the data that is transferred. Because of the platform-neutral nature of XML, SOAP, and WSDL, Java is a popular choice for web-service implementation due to its OS-neutrality.

There are key issues which need to be address in RESTful based SOA; 1. Object in REST is represented by a URL and it is responsible for including state information in each message, so object processing is always stateless but complex objects which represents multi-stage processing is difficult to manage in RESTful web services. 2. Key feature of SOAP-based SOA is level of discovery and binding flexibility. It is easy for producer to register SOAP web services and easy to search it from registry using UDDI because of standard and industry accepted registry. But in case of RESTful web service, no such industry accepted central registry exists which is prime concern of this research paper. Some researcher proposed hybrid approach which uses REST at the front-end portion of the application and SOAP at the back-end [10]. Focus of this research paper is to address this issue and provide optimal solution 3. Security of RESTful interface is also key concern because in SOAP-based SOA, developer has explicit control over security [6, 11]. 4. Orchestration and workflow threading is prime concern of architect for RESTbased SOA. 4. There is no intermediary specific mechanism exist to ensure failure of RESTful web service which can be backtrack [8].

Recently RESTful web service gained momentum in development of distributed application because it offers faster path to develop, easy to integrate with web application, handheld devices, mobile application and it is best solution for many SOA application [9].

The remainder of the paper is organized as follows: Section 2.0 discusses recent research work which provides the foundation for our research. Section 3.0 describes Travel Scheduling

motivating scenario which will be used to develop benchmark application. Section 4.0 Discuss Proposed framework: XML based Registry for RESTful web service: RESTfulRegistry. 5.0 describes benchmark application configuration to develop benchmark application and results. We conclude our research work and presents future research directions in Section 6.0.

## 2. Related Work

During last decade following researchers contributed in composition of RESTful web service.

Qing Liu, Chenhe Liu, Huian Li, Xiang Xu and Lexi Gao introduced technique to discovery real world RESTful web service from world wide web. They developed RESTWS tool by using Heritrix and some other open source tools to crawl the web page to find candidate RESTful web services[3]. Author suggested technique to search RESTful web service from World Wide Web but we don't discuss ways to registry it in standard registry for future use.

Tiemei Irene Zhang and Janaka Indrajith Wijayanayake proposed framework for REST service registry. Proposed framework uses oracle database to store services but we don't find any concrete implementation or prototype[4].

## 3. Motivating Scenario

We used travel scheduling motivating scenario to describe our proposed approach. In this scenario customer wants to travel from departure location to destination location with hotel booking. It is time consuming and tedious job for customer to find different ways like traveling by bus / by train to intermediate destination and also finding travel option from intermediate destination to goal city with hotel booking. There are many RESTful web services are available which will provide travel information (either by bus or by train or by airplane), hotel information and payment information but these individual services are not enough to give complete travel plan. This is perfect example where travel schedule plan with hotel booking and payment can be generated automatically based on customer need. We developed benchmark application based on proposed framework where consumer can discover RESTful web service based on their requirement.

## 4. Proposed Framework

XML based Registry for RESTful web service - RESTRegistry

In this research paper we proposed framework has three component: 1. Service Provider can register service 2. Service Consumer can discover service based on requirement 3. Repository store services with quality of service factor. Following definitions provide foundation for RESTRegistry.

Definition 1: (Web Service) Web service can be defined with 2-tuple  $WS = (Iw, Ow)$  where  $Iw = \{Iw_1, Iw_2, \dots\}$  is an input parameter set and  $Iw_i \in Iw$ .  $Ow = \{Ow_1, Ow_2, \dots\}$  is an output parameter set and  $Ow_i \in Ow$  [5]. We used following web services (shown in Table. 1.0) in our motivating scenario.

Table 1: Service name with specification

Service Name	Specification
ANLineService	Calendar Get_Arrival_TimeOfAN(int hour, int minute) String get_Train_NameforAN(int hour, int minute)
JetAirwaysService	Calendar Get_Arrival_TimeOfAirline (int hour, int minute) String get_Airline_NameforNtoL(int hour, int minute)
LondonLutonAirportBusService	Calendar Get_Arrival_TimeOfBus (int hour, int minute) String get_Bus_NameforLtoL(int hour, int minute)
LutonAirportGoalAirportBusService	Calendar Get_Arrival_TimeOfBusforLtoG (int hour, int minute) String get_Bus_NameforLtoG(int hour, int minute)
GoalAirportGoalCityBusService	Calendar Get_Arrival_TimeOfBusforGtoG (int hour, int minute) String get_Bus_NameforGtoG(int hour, int minute)
EmiratesService	Calendar Get_Arrival_TimeOfAirlineforNtoL (int hour, int minute) String get_Airline_NameforNtoL(int hour, int minute)
AirIndiaService	Calendar Get_Arrival_TimeOfAirlineforAtoL (int hour, int minute) String get_Airline_NameforAtoL(int hour, int minute)
MacHotelService	String[] Get_Mac_Hotel_Service ()
MacRoomService	String[] Get_Mac_Room_Service ()
MacAmenityService	String[] Get_Mac_Amenity_Service ()
MacProvinceService	String[] Get_Mac_Province_Service ()
MacBookingService	String[] Get_Mac_Booking_Service ()
CherryHotelService	String[] Get_Cherry_Hotel_Service ()
CherryRoomService	String[] Get_Cherry_Room_Service ()
CherryAmenityService	String[] Get_Cherry_Amenity_Service ()
CherryProvinceService	String[] Get_Cherry_Province_Service ()
CherryBookingService	String[] Get_Cherry_Booking_Service ()
TravelReserveService	String[] do_Reservation(String[] service_id[])
TravelPaymentService	String[] do_Payment(String[] service_id[])

Definition 2: (Service Type) Service type is defined with 1-tuple service type  $St1 = \{ws1\}$   $St2 = \{ws2\} \dots$  where  $ws1, ws2 \dots$  is instance of service. We developed following (Table. 2.0) Service type in our motivation scenario.

Definition 3: (Service URL) Service URL defined as with 1-tuple; Service URL  $SU1 = \{wsurl1\}$ ,  $SU2 = \{wsurl2\} \dots$  where  $wsurl1, wsurl2 \dots$  is URL of resources of RESTful web services.

Table 2: Detail of Service type with service name used in motivating scenario

Service Type	Service Name
move ahmedabad newdelhi	ANShatabdiTrainService
move newdelhi london	JetAirwaysService
move london luton_airport	LondonLutonAirportBusService
move luton_airport goal_airport	LutonAirportGoalAirportBusService
move goal_airport goal_city	GoalAirportGoalCityBusService
move newdelhi luton_airport	EmiratesService
move ahmedabad london	AirIndiaService
avail mac_hotel_service	MacHotelService
avail mac_room_service	MacRoomService
avail mac_amenity_service	MacAmenityService
avail mac_province_service	MacProvinceService
avail mac_booking_service	MacBookingService

Definition 4: (Quality of Service) Quality of Service is defined 5-tuple  $QOS = \{qosf1, qosf2, \dots\}$  where  $qosf1, qosf2, \dots$  is a quality of service factor. In our motivational scenario Service Level Measurement Quality (SLMQ), Business Processing Quality (BPQ), Manageability Quality (MQ) is quality of service factor.

Definition 5: (Quality of Service Sub Factor) is defined as  $QOSSF = \{\{qossf1, qossf2, \dots\}, \{qossf1, qossf2, \dots\}\}$  where  $\{qossf1, qossf2, \dots\} \in qosf1$  and  $\{qossf1, qossf2, \dots\} \in qosf2$ . In our motivating scenario we consider response time, throughput, availability, accessibility success ability as sub quality factor of service level measurement quality; service recognition, messaging reliability, transaction integrity, collaborability as sub quality factor of business processing quality; informability, observability, controllability as sub quality factor of manageability quality.

Definition 6: (Web Service Repository) Web service repository is defined as  $WSR = \{St1, Ws1, Qos1, St2, Ws2, Qos2, \dots\}$  where  $St1, St2, \dots$  is service type;  $Ws1, Ws2, \dots$  is instance of web service;  $Qos1, Qos2, \dots$  is quality of service factor.

Proposed framework is show in Fig. 1.0 where service provide can register RESTful web services and service consumer can discovery it based on their requirement.

## 5. Benchmark Application Configuration and Results

We developed benchmark application using JSP and XML used to create RESTfulRegisgry. Restful Web Service developed using java (JSON format data) with Asyn receiver. Android Mobile client developed using Android SDK API 2.2. and Apache Jmeter used to measure performance of the application. Travel scheduling web application deployed on Apache web server. Web application configured and results measured on HP ProBook 4530s with Intel( R ) Core ( TM) i3 -2310M CPU @ 2.10 GHz, 4.00 RAM and 64-bit Operating System. Output of benchmark application is shown in Fig. 2.0.

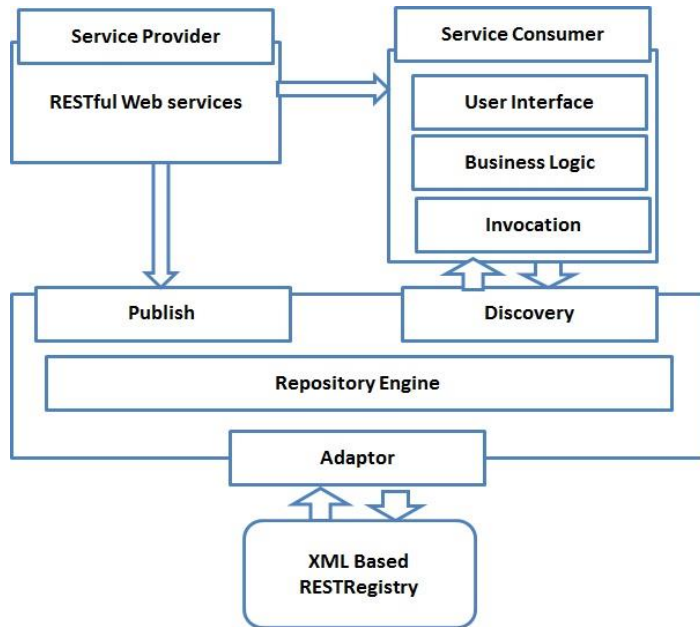


Figure 1: Framework for XML based Registry for RESTful web service

```

<service_type>
<service>
<service_name
attribute="move">ahmedabad
newdelhi</service_name>
<service_url> https://machotel.com/
ahmedabad newdelhi</service_url>
<quality_factor><slmq>
<responsetime> 04 </responsetime>
<throughput> 05 </throughput>
<availability> 06 </availability>
<accessibility> 07 </accessibility>
<successability> 05 </successability>
</slmq>
<bpq>
<servicerecognition>01</servicerecognition
>
<messagingreliability>02</messagingreliabil
ity>
<servicerecognition>01</servicerecognition
>
<messagingreliability>02</messagingreliabil
ity>

```

## 6. Conclusion

In this research paper we proposed framework with XML based repository: RESTfulRepository through which service producer can easily register RESTful web service and server consumer can discover service based on their specification. We also consider quality of service with sub quality factor which is can update regularly for each RESTful web service. RESTfulRegistry developed using XML technology which is compatible with any platform and programming language and we also describe implementation of proposed framework using Travel scheduling scenario and tested in real word. Proposed framework provide foundation for development of dynamic composition of RESTful web service and RESTbased SOA.

In future we will work on searching of RESTful web services based on quality of service and bind it dynamically with web based client and also try to use it in more complex scenario such as workflow generation or composition of RESTful web service to generate value added services.

## References

- [1] Refe R. T. Fielding. Architectural Styles and the Design of Network-based Software Architectures. PhD thesis, University of California, Irvine, 2000,pp. 89-88.
- [2] Rathod D.M, Parikh, S.M. and Buddhadev. B.V. “Structural and behavioral modeling of RESTful web service interface using UML,” IEEE International Conference on Intelligent Systems and Signal Processing (ISSP), 2013, pp. 28-33 DOI: 10.1109/ISSP.2013.6526869, Print ISBN: 978-1-4799-0316-0
- [3] Qing Liu, Chenhe Liu ; Huian Li ; Xiang Xu ; Lexi Gao , “Towards Automatic Discovering for a Real-World RESTful Web Service”, in Proc of 9th IEEE International conference on Web Information Systems and Applications Conference (WISA), 2012, Date 16-18 Nov. 2012, pp. 39-42
- [4] Tiemei Irene Zhang and Janaka Indrajith Wijayanayake, “A Framework for REST Service Registry”, 8th International Conference on Computing, Communications and Control Technologies: CCCT 2010, April 6 th - 9 th, 2010 ~ Orlando, Florida USA.
- [5] Rauf, A. Ruokonen, T. Systa, and I. Porres, "Modeling a composition RESTful web service with UML," in Proc of the 4th European Conference on Software Architecture (ECSA 2010), Copenhagen, 2010, pp. 253-260.
- [6] H. Zhao and P. Doshi, "Towards automated RESTful web service composition," in Proc of the 2009 IEEE International Conference on Web Services (ICWS '09), Washington, 2009, pp. 189-196.
- [7] Agarwal, V.—Chafle, G.—Mittal, S.—Srivastava, B.: Understanding Ap-proaches for Web Service Composition and Execution. In COMPUTE'08: Proceed-ings of the 1st Bangalore Annual Compute Conference, New York, NY, USA, ACM 2008, pp. 1–8.
- [8] Alrifai, M.—Risse, T.—Dolog, P.—Nejdl, W.: A Scalable Approach for Qos-Based Web Service Selection. In Service-Oriented Computing 2008 Workshops, Springer-Verlag, 2009, pp. 190–199.
- [9] Bartalos, P.: Effective Automatic Dynamic Semantic Web Service Composition. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 3, 2011, No. 1, pp. 61–72. [1].
- [10] Bellur, U.—Vadodaria, H.: On Extending Semantic Matchmaking to Include Preconditions and Effects. In ICWS '08: Proceedings of the 2008 IEEE International Conference on Web Services, Washington, DC (USA), 2008, pp. 120–128.
- [11] DiBernardo, M.—Pottinger, R.—Wilkinson, M.: Semi-Automatic Web Ser-vice Composition for the Life Sciences Using the Biomoby Semantic Web Framework. Journal of Biomedical Informatics, Vol. 41, Issue 5, Semantic Mashup of Biomedical Data, ISSN 1532-0464, 2008, pp. 837–847.
- [12] Rakesh Bhatnagar, Dr. Jayesh Patel, “Scady: A Scalable & Dynamic Toolkit for Enhanced Performance in Grid Computing”, IEEE International Conference on Pervasive Computing ICPC - 2015, Jan 8-10, 2015, Pune.
- [13] Rakesh Bhatnagar, Dr. Jayesh Patel, “API Specification for Small Grid Middleware - Scady”, IEEE Xplore, India Conference (INDICON), 2014 Annual IEEE, DOI:10.1109/INDICON.2014.7030545 Publication Year: 2014, Page(s): 1 - 5.
- [14] Digvijaysinh M. Rathod, M S Dahiya, Satyen M. Parikh, “Towards composition of RESTful web services”, 2015 IEEE 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 13-15 July 2015, DOI: 10.1109/ICCCNT.2015.7395237.

- [15] Digvijaysinh M Rathod, Satyen M Parikh, BV Buddhadev, “Structural and behavioral modeling of RESTful web service interface using UML” 2013, IEEE International Conference Intelligent Systems and Signal Processing (ISSP), 2013, 1 March 2013, DOI: 10.1109/ISSP.2013.6526869.

---

\*Corresponding author.

*E-mail address:* digvijay.rathod@ gfsu.edu.in