

# Mobile Integration Using Oracle Service Bus



Oracle Service Bus 12c enables enterprises to deliver on mobile as an extension of the integration platform. Developers can create REST/JSON APIs and simplify the process of creating customizable applications from reusable components. Oracle Service Bus 12c makes it easy to extend on-premise and cloud applications to the mobile channel.

## KEY FEATURES

- Mobile enablement with REST/JSON support
- Unify Cloud and On-Premise applications and services
- Unify with Oracle Mobile Application Framework for mobile front-end development
- Built-in caching to handle high volumes of mobile users
- Intelligent content and identity based routing
- Built in monitoring, management, and QoS

## KEY BENEFITS

- Simplify integration and reduce costs
- Unify framework across cloud, mobile and on-premise
- Extreme performance and scalability
- Improve developer productivity
- Reduce support and maintenance costs
- Faster time to market for mobile services

## Introduction

For many years web browsers and desktop computers were the primary way to access information from enterprise applications. As mobile devices have proliferated, applications are no longer tied to the desktop. Users want to use their smart phones and tablets to access corporate data and business apps, anytime, anywhere. Many IT departments are having a hard time accommodating these mobile interfaces while preserving hard-won enterprise standards. Rather than continuing to develop applications first for the desktop and then making tactical mobile development choices, IT leaders want a consistent architecture that considers all channels.

Many Oracle customers are addressing this challenge. Rather than treating mobile as a entirely new development and deployment channel, they see mobile integration as an extension of the SOA integration methods that they already have in place.

## Mobile Integration with Oracle Service Bus

The Oracle Mobile Suite utilizes standard technologies and tools to expose many data formats for exchange data and functions with any mobile application and includes Oracle Service Bus and Oracle Mobile Application Framework. Oracle Service Bus supports all types of connections between applications on mobile devices and back end business systems including the popular REST/JSON.

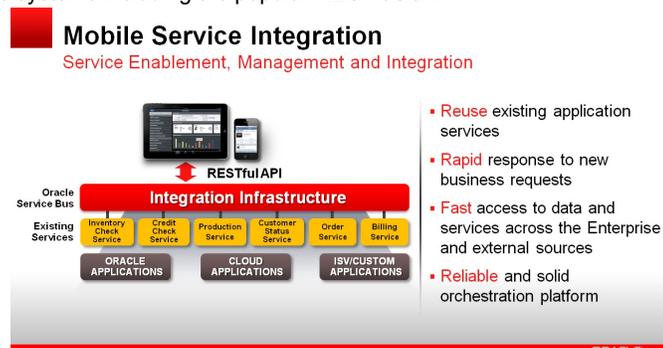
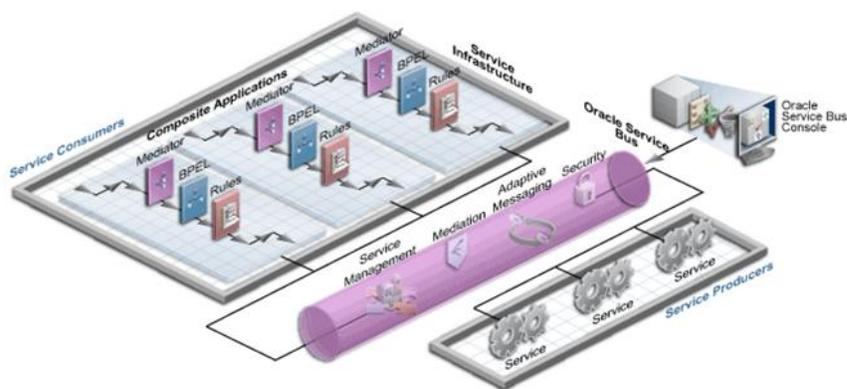


Figure 1 Mobile Service Integration

Mobile applications communicate with back-end applications through a standardized mediation and virtualization layer—Oracle Service Bus—which can connect to the backend web services and expose them as RESTful APIs. This allows developers to leverage their existing application and integration infrastructure including applications hosted on the cloud as part of a highly reusable and scalable development and integration platform. With this, mobile applications don't have to be self-contained and data doesn't have to be stored on the device.

### Oracle Service Bus

Oracle Service Bus is a scalable and reliable Enterprise Service Bus. It is designed to connect, virtualize and manage interaction between heterogeneous services, legacy systems, enterprise systems, mobile devices, the cloud, and other service bus instances across the enterprise. It is more than a mere messaging engine; it is the communications medium that connects all assets of the enterprise SOA and provides industry standard interfaces to the enterprise as a whole, not just to specific applications.



#### ORACLE SERVICE BUS FOR ENTERPRISE MOBILITY

#### RELATED PRODUCTS

Oracle Service Bus complements the Oracle Fusion Middleware platform as a very well integrated, high performing Enterprise Service Bus.

Oracle Service Bus is also included in:

- Oracle SOA Suite
- Oracle Mobile Suite

Figure 2 Oracle Service Bus – Architectural Position

The key to the value of Oracle Service Bus in any architecture is the concept of, “virtualization”. The Oracle Service Bus can translate and transform message formats, integration technologies and security schemes to provide the loose coupling between systems. By mediating the interactions of disparate systems, the Oracle Service Bus provides much of the promised agility of SOA.

#### Key Benefits

- » High availability —Delivers stringent SLAs with intelligent load balancing and in-built high-availability infrastructure
- » Resource optimization—Minimizes costs by maximizing utilization of existing server resources even at low-latency, high-volume workloads
- » Ease of management—Allows instant, point-in-time, restore for configuration settings and provides anytime, anywhere accessibility with a 100% web-based environment
- » Built in caching – Handles high volumes of mobile users for enhanced end-

user experience.

## Product Overview

Oracle Service Bus is a part of Oracle SOA Suite 12c. It introduces a REST binding within JDeveloper to simplify mobile enablement by exposing traditional SOAP services, Enterprise Java Beans (EJBs), JCA adapters connecting backend applications or just about any other underlying implementation through REST/JSON.

The REST binding is available for SOA composites and Oracle Service Bus services and allows the configuration of REST interactions as exposed service or proxy service. It also allows the invocation of externally available REST services.

Multiple ways to define a REST binding through a simple wizard are included: The definition can be on an existing WSDL, or the binding can even create the WSDL for you. When using the REST binding, a WADL (Web Application Description Language) file is created automatically to define the REST interaction and to allow interaction with the REST binding throughout the integration layer.

SOA Suite 12c practices a “REST on the edges” approach where a REST/XML, REST/JSON, or URL-encoded payload is translated to and from SOAP/XML in the integration wire. This is called a “translation on the wire”. This approach has been selected because SOAP/XML has been well accepted for many years. Any experienced integration developer is quite familiar with SOAP/XML interactions and will likely want to reuse as much as possible rather than a complete “rip-and-replace”. By translating, you can interact with REST style edge interfaces while leveraging your existing integration layer. Using this allows us to very easily expose an existing SOAP/XML interface as a REST style interface.

By exposing back-end applications as RESTful APIs through your existing SOA composite or Service Bus, which then connects to backend services and applications, you are able to leverage your existing application and integration infrastructure as part of a highly reusable and scalable development and integration platform while your mobile applications communicate through Oracle Service Bus as standardized mediation and virtualization layer.

The screenshot below shows a Service Bus pipeline, calling a backend SOAP service and exposing a REST and SOAP interface.

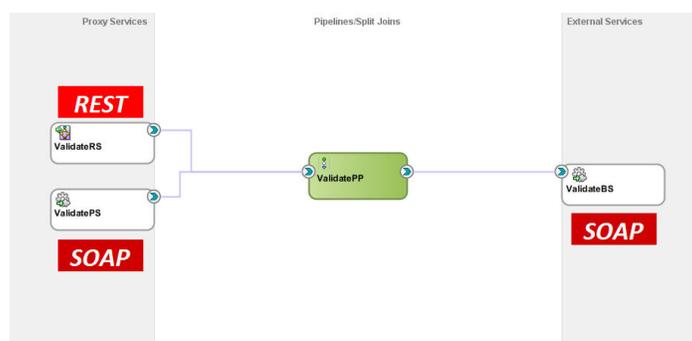


Figure 3 Oracle Service Bus exposing a backend SOA service as REST service

The REST binding wizard is exactly the same for SOA composites and for Service Bus projects. The remainder of this section will discuss REST support within Service Bus to avoid duplication. When exposing an existing SOAP service, the REST binding automatically takes the same shape as defined in the original WSDL. If a different shape needs to be supported, a REST interface can be created based on a different schema and messages can be transformed in the pipeline. The translation (REST to SOAP to REST) takes place on the wire between the pipeline and the Proxy and/or Business services.

The screenshot below shows how a SOAP based payment service is exposed as REST service through the new wizard. You can see that the parameters for the REST service are auto-created and can manually be changed.

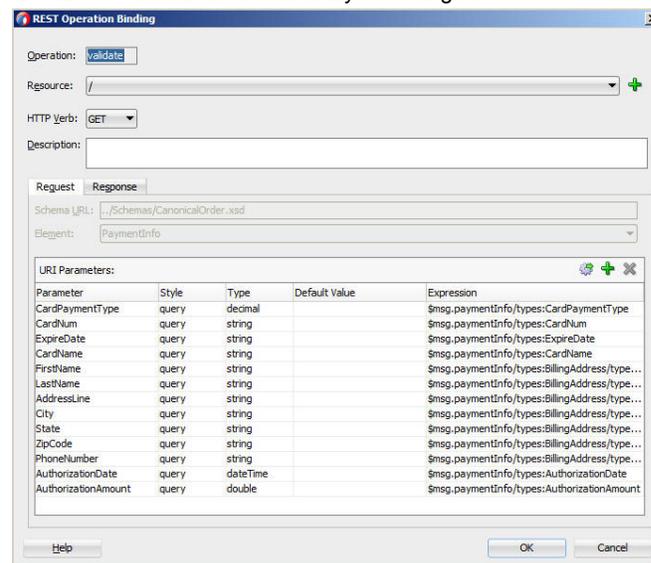


Figure 4 REST binding wizard

The wizard allows the definition of the payload type as XML or JSON or both, create sample data as XML or JSON and to define HTTP status for each operation.

The HTTP Verb dropdown allows the creation of bindings for all of the standard verbs. The Service Bus test console has been enriched for easy testing of REST services by providing an input field for the resource to be tested and drop-down menus to select the method and the accept header.

## Summary

Based on time-tested and sound SOA principles, Oracle Service Bus can extend existing and new enterprise applications to develop a mobile channel. As part of the Oracle Mobile Suite, Oracle Service Bus delivers the integration and mediation layer to REST/JSON enable services to expose them to build mobile applications, while Oracle Mobile Application Framework can be used to deliver on the front-end application development independent of the underlying operating system. Oracle Service Bus makes it possible to leverage cloud and on-premise applications to build out a new mobile channel that delivers on increased business value and customer service.

**CONTACT US**

For more information about Oracle Service Bus for Mobile Integration, visit [oracle.com/soa](http://oracle.com/soa) or call +1.800.ORACLE1 to speak to an Oracle representative.

**CONNECT WITH US**

[blogs.oracle.com/soa](http://blogs.oracle.com/soa)



[facebook.com/oraclesoa](https://facebook.com/oraclesoa)



[twitter.com/oraclesoa](https://twitter.com/oraclesoa)



[oracle.com/soa](http://oracle.com/soa)

**Hardware and Software, Engineered to Work Together**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0614

