SOA rocks!

Adventures of a service oriented architect

☰ Menu

SOAP is dead – Lessons Learned from SOA-fying a Monolith

2015-10-10 by Stefan Macke

I’ll continue my series of blog posts regarding the lessons we learned while SOA-fying our monolithic Adabas/Natural application with a more technical lesson:

> *SOAP is dead.*

bykst / https://pixabay.com/de/speckstein-stein-grabstein-material-447299/

This may be a harsh statement, taking into account that we started out with Webservices based on SOAP and at the moment our whole infrastructure is based on it.

However, while we were still searching for the right solution to the communication problem two major programming languages stopped supporting SOAP out of the box: Groovy and Ruby. And we used both of them.

If you take a closer look at the current notions in architecture and distributed systems, you'll find plenty of conference talks and frameworks for providing and consuming REST services. However, SOAP seems to be legacy technology already.

To be honest, we don't have a single REST service in production, yet. But after working with SOAP Webservices for quite some time now, I can definitely see the advantages of a more loosely coupled approach like REST. The interface design for SOAP services can take quite a lot of time, because you have to define individual operations and data types. And changing the interface – e.g. adding a parameter – leads to a required re-build of all systems using it. With REST you only need simple CRUD operations and can use JSON as a very flexible data format that can be added to later on.

I'm glad that Integration Server provides an easy means to publish REST services. You can even put them on top of existing services and simply provide a more flexible interface to the consumers. This is definitely a solution I want to try out in the near future.

So, if you start with an SOA today, take a closer look at REST and don't blindly use the default implementation SOAP as it may not be supported any more in the near future.

Blog

Lessons Learned, REST, SOAP, Web Services

‹ Test everything! – Lessons Learned from SOA-fying a Monolith

› Avoid Redundancy! – Lessons Learned from SOA-fying a Monolith

3 thoughts on "SOAP is dead – Lessons Learned from SOA-fying a Monolith"

Mubeen

2017-03-07 at 6:28:51

Soap also requires more learning than REST. SOAP in my opinion is complex. This WSDL, things seems difficult to absorb therefore YES! I think SOAP is now legacy

Valentin Bojinov

2017-10-28 at 18:01:33

Read about BPM and bowl before sending soap/wsdl to hell.

Stefan Macke

2017-11-04 at 15:30:38

Hi Valentin,

I’m using SOAP myself 😉 The image was only meant to provoke thinking about the pros and cons of the technology.

Best regards,
Stefan

Leave a Comment

B *I* “ { } ?

Name *

Email *

Website

□By using this form you agree with the storage and handling of your data by this website. *

Post Comment

About me

Hi, I'm Stefan Macke.

I'm a software developer and architect at ALTE OLDENBURGER in Germany where I'm working with Java, .NET, Adabas/Natural, and webMethods Integration Server.

Search …

Legacy Coder Newsletter

My newsletter keeps you up to date with the podcast. It might also contain information about my events, products, offers, and promotions. You can read more about privacy, cancellation, logging, and tracking in my Privacy Policy (German).

Subscribe to the Legacy Coder Newsletter now!

Google Ads

Meta

- Log in
- Entries feed
- Comments feed
- WordPress.org

Recent Posts

WFLYCTL0348: TimeoutException while running Keycloak in a Docker container with an external database (MariaDB)

German Natural User Group Meeting in November 2018 – Legacy Coder Podcast #7

Continuous Deployment with Natural – Legacy Coder Podcast #6

What is Legacy Code? – Legacy Coder Podcast #5

How to deploy to JBoss EAP 7 with Gradle, Cargo, and Jenkins

Recent Comments

Stefan Macke on How to create a REST service in webMethods Integration Server

TC on How to create a REST service in webMethods Integration Server

Asmaâ on How to create a new webMethods Integration Server instance

#2 (IF #Standards = true AND #Good-practice = false) THEN Software Conventions := 'Virtual'? | Mr Natural One on German Natural User Group Meeting in November 2018 – Legacy Coder Podcast #7

Ali Jahanzaib on Logging to webMethod's Integration Server log from a Java Service

Tags

Adabas Administration Application Server Arquillian Best Practices Build Continuous Integration Conventions Databases Deployment Domain Specific Languages Errors Git Gradle HowTos HTTPS Java Java EE Java EE 7 JBoss Jenkins JMS Lessons Learned Logging Microservices Modernization Natural NaturalONE Performance Programming REST Ruby Security SOA-fication SOAP Soft Skills Software AG Natural Talks Teamwork Testing UnitTests User Groups webMethods Integration Server Web Services Wildfly