

Building Applications Programming Interfaces for the Future

Tuesday, January 26, 2016

NY Software Process Improvement Network
ThoughtWorks
New York, New York

Robert Gezelter Software Consultant

Introduction

APIs are Forever

“Alphabet Soup”

Syntax vs. Semantics

The Support Problem

. . .

- APIs are Forever
- “Alphabet Soup”
- Syntax vs. Semantics
- The Support Problem
- Identifying Key Issues
- Publishing Definitions
- Validation/Sandboxes
- The Future

APIs are Forever

- APIs are export products
- Outsiders will invest \$+++ in code
- Deprecation perhaps; Removal unlikely
- Shortcomings have LARGE consequences

Robert Gezelter Software Consultant

Introduction

APIs are Forever

“Alphabet Soup”

Syntax vs. Semantics

The Support Problem

. . .

“Alphabet Soup”

REST

Asynchronous

HTTP

XML

SOAP

JSON

Synchronous

MultiRequest

Syntax vs. Semantics

- The “Mad Libs(TM)” Test
- JSON vs. XML – irrelevant
- JSON with eval() – dangerous
- RPC vs. REST – relevant
- SOAP vs. REST – irrelevant

Robert Gezelter Software Consultant

Introduction

APIs are Forever

“Alphabet Soup”

Syntax vs. Semantics

The Support Problem

. . .

JSON compared to XML

```
{ "FirstName" : "Robert",  
  "LastName" : "Gezelter",  
  "Address" : {  
    "StreetAddress" : "3520 167th Street",  
    "City" : "Flushing",  
    "State" : "NY",  
    "PostalCode" : "113581731"},  
  "PhoneNumbers" : [  
    {"type" : "office"  
     "Number" : "7184631079"},  
    {"type" : "mobile"  
     "Number" : "....."}]  
  "LineItems" : [{  
    ... }]  
}
```

Robert Gezelter Software Consultant

Introduction

APIs are Forever

“Alphabet Soup”

Syntax vs. Semantics

The Support Problem

. . .

JSON compared to XML

```
<Order>
  <FirstName>Robert</FirstName>
  <LastName>Gezelter</LastName>
  <Address>
    <StreetAddress>3520 167th Street</StreetAddress>
    <City>Flushing</City>
    <State>NY</State>
    <PostalCode>113581731</PostalCode>
  <PhoneNumbers>
    <type>office</type>
    <Number>7184631079</Number>
    <type>mobile</type>
    <Number>.....</Number>  </PhoneNumbers>
  <LineItems>
    ...  </LineItems>
</Order>
```

The Support Problem

- Bugs are expensive
- Ambiguity is worse
- Problems in the age of Twitter
- Whose fault?

Identifying Key Issues

- Synchronous/Asynchronous
- Transport (e.g., HTTP, SMTP)
- Integrity
- Security (e.g., HTTPS, HTTP)
- Expandability
 - Transaction IDs
 - e.g., HTTP 1.0 vs. HTTP 1.1
 - Expandability mitigates obsolescence

Publishing Definitions

- Imprecision causes confusion
- Less expensive than later problems
- Schemas provide a formalism
- Grammars beyond schemas
- State machines
- Character sets (e.g., ASCII, UTF-8)

Validation/Sandboxes

- Publish good examples
- Maximize use of sandboxes
- Grammar+Schema implies correctness
- Use same tools for testing

The Future

- Well-thought out APIs are timeless
- Less diversity; less expense
- Preserve customer code investment
- Avoid error resolution

Summary

APIs should NOT be considered agile, they are a contract with your customer and future. Good specifications reduce long-term, aggregate costs for vendor and consumer.

Questions?

Robert Gezelter Software Consultant
35 – 20 167th Street, Suite 215
Flushing, New York 11358 – 1731
United States of America

+1 (718) 463 1079
gezelter@rlgsc.com
<http://www.rlgsc.com>

Session Notes & Materials:

<http://www.rlgsc.com/nyspin/2016-01/building-apis-for-the-future.html>