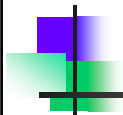




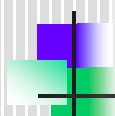
～ 第8回 XMLコンソーシアムDay ～
Web2.0部会活動中間報告

RESTとSOAP

メリット比較, SOAP記述からのREST自動生成,
WADL, WSDL2.0によるRESTの動向, SOAPを"乱暴に使う"



2006年12月11日
XMLコンソーシアム Web2.0部会
荒本道隆(アドソル日進株式会社)



アジェンダ



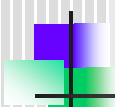
- はじめに
- 概要
- サービスの定義方法
- 特徴
- スクリプト言語からの利用
- SOAPをREST対応にする方法
- マッシュアップ
- 今後の動向とまとめ



はじめに

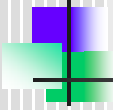


- かつてWebサービスと言えはSOAPでしたが、現状ではInternet上で公開されているWebサービスはRESTが多くなってきています。
- そのような現実を踏まえ、SOAPとRESTについて「どう違うのか?」「どっちを使ったらいいんだ?」という疑問について、部内で検討した内容を報告します。



REST と SOAP

概要

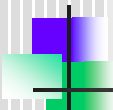


SOAPとは



XML Consortium

- Simple Object Access Protocol
 - SOAPを使ったWebサービス
- リクエストのXMLを送ると、レスポンスのXMLが返ってくる
- SOAP 1.0
 - SOAP1.0 = Webサービス
- SOAP1.1
 - HTTP以外にも利用可能に
 - SMTP, FTP, e.t.c...
- SOAP1.2
 - GETが利用可能に
 - <http://www.w3.org/TR/soap12-part0/#L26854>



SOAPのサンプル



XML Consortium

```

SOAP 1.1
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <Header>
    <GetHello/>
  </Header>
  <Body>
    <GetHello/>
  </Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Disposition: inline

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <Header>
    <GetHelloResponse/>
  </Header>
  <Body>
    <GetHelloResponse/>
  </Body>
</soap:Envelope>
  
```

```

@WebServiceNamespace xmlns="http://www.xml.com/..."
@WebServiceTargetNamespace xmlns:tns="http://www.xml.com/..."
Public Class WebService
  Inherits System.Web.Services.WebService

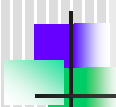
  <WebServiceMethod> ...
  Public Function GetHello(ByVal str As String) As String
    Return "Hello World"
  End Function
End Class
  
```



RESTとSOAPの違い



- REST
 - 1つのURLに対して、1つの情報を取得できる
 - URLの?以降のパラメータも、URLの一部
例 . <http://hostname/hoge.cgi?id=1234>
<http://hostname/hoge.cgi/id/1234/>
- SOAP
 - エンドポイントに対して、XMLを送ると、結果のXMLが返ってくる
 - Document型



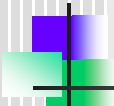
REST と SOAP サービスの定義方法



RESTの定義方法



- WADL
 - Web Application Description Language
 - <https://wadl.dev.java.net/>
 - RelaxNG Schema か W3C XML Schema で記述する
 - DELETE, GET, HEAD, POST, PUT
 - 20061109 - Specification (PDF) ¹B RelaxNG Schema for WADL₂ より
 - 対応しているミドルウェアはまだ少ない
 - wadl2java



WADLサンプル



```

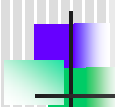
<?xml version="1.0" ?>
<!-- application:xml:uri="http://www.w3.org/2001/XMLSchema-instance"
xml:base="http://www.w3.org/2001/XMLSchema-instance"
xmlns:foo="http://www.w3.org/2001/XMLSchema-instance"
xmlns:bar="http://www.w3.org/2001/XMLSchema-instance"
xmlns:baz="http://www.w3.org/2001/XMLSchema-instance"
xmlns:qux="http://www.w3.org/2001/XMLSchema-instance" -->
<!-- schema: -->
<!-- include href="http://www.w3.org/2001/XMLSchema-instance" -->
<!-- schema: -->
<!-- resources base="http://api.search.yahoo.com/services/V1/" -->
<!-- resource path="search" -->
<!-- method name="GET" id="search" -->
<!-- request -->
<!-- response -->
<!-- media type -->
<!-- error -->
</resources>
</application>

```




REST と SOAP

特徴



RESTの特徴

- 利用し易い
 - HTTPリクエストが使える環境であれば、呼び出す事ができる
 - パラメータは、文字列連結するだけ
 - レスポンスのXMLから必要な部分だけを取り出す
 - ブラウザを使って、動作テストができる
 - 簡単にサーバの挙動を確認できる
 - **使ってみようという気になる**
 - 検索のWebサービスは、RESTが非常に多い

RESTの特徴



- 開発し易い
 - サーバ側は、Webサイト開発のノウハウがそのまま使える
 - Servlet: `request.getParameter("param1");`
 - cgi: `QUERY_STRING`
 - Webサイトを実装できる環境であれば、何でもOK
 - クライアント側は、HTTPのリクエストを発行
 - JavaScript `XMLHttpRequest`, `ActiveXObject("Microsoft.XMLHTTP")`
 - Java `URLConnection`
- HTTPの特徴をそのまま利用できる
 - httpsによる暗号化
 - SessionやCookieの利用
 - RESTfulではなくなる
 - URLの長さ制限
 - 仕様上の制限は無いが、技術的な制限があるので、あまり長いURLは使えない
 - POSTの利用を検討

RESTに適しているサービス



- 入力パラメータが少ない
 - WS-* を必要としない
- ↓
- 不特定多数を対象とした検索サービス等**
- Google, Amazon, Yahoo などの検索サービス
-
- 使い方次第で、ブラウザのドメイン境界問題を回避できる
- ↓
- HTMLの<xxx src="...">を書き換える**
- GoogleMapsなど

SOAPの特徴



- WS-*による拡張性
 - WS-Security
 - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
 - XML署名, XML暗号化により、完全性や秘匿性を実現
 - End-to-Endのセキュリティ
 - WS-Policy
 - <http://www.w3.org/2002/ws/policy/>
 - Webサービスに関するポリシーを記述
 - WS-Addressing
 - <http://www.w3.org/2002/ws/addr/>
 - Webサービスのエンドポイントを参照
 - WS-.....
 - 仕様検討中を含めて、まだまだ一杯あります
 - W3CやOASISなどの標準化団体により標準化

SOAPの特徴



- リクエスト時に、複雑な情報を送れる
 - リクエストもXMLなので、構造化された情報を送れる
 - 複数のXMLを1つのXMLに入れる
- スキーマによる厳密なバリデーション
 - 入出力パラメータのフォーマットチェック
 - XML名前空間による厳密性
- SOAP対応のミドルウェア
 - Java(axis, JAX-WS)
 - .NET Framework
 - PHP(SOAP拡張モジュール)
 - Perl (Perl/SOAP)
 - データベース (Oracle, DB2, SQLServer)
 - MS-Office (SOAP Toolkit)

SOAPに適しているサービス



- 入力に対しても、スキーマを持っている
 - 入力が複雑な構造を持っている
 - 入力を厳密にチェックする必要がある



基幹システム, B2Bなど

- WS-* を必要とするもの



WS-Securityを使った例 sPlat

XML Consortium

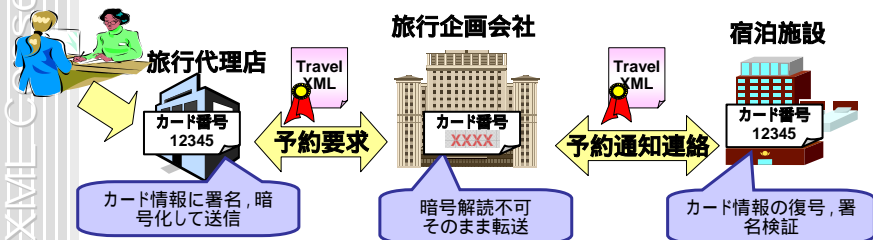
参考資料: sPlatの概要



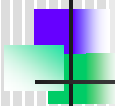
Webサービス実証部会とセキュリティ部会の共同プロジェクト

- 中継者を介したマルチホップのWebサービス(SOAP)
- WS-SecurityによるEnd-to-Endのセキュリティを確保
 - httpsによるトランスポート層の暗号化では、中継者のところで一旦すべて復号されてしまう
 - WS-Securityによるドキュメント層の暗号化を使えば、中継者に見せたくない情報を見えないように出来る
- sPlatでは、中継者の具体的な課題の解決策を検討中

利用者



XML Consortium



REST と SOAP

スクリプト言語からの利用



スクリプト言語でのREST,SOAPの利用

- JavaScript
 - Ajax(Asynchronous JavaScript + XML)
 - XMLHttpRequest
 - ブラウザのJavaScriptからRESTやSOAPを利用する
- PHP, Perl, e.t.c....
 - PHP5からSOAP拡張モジュールが利用可能になった
 - <http://php.morva.net/manual/ja/ref.soap.php>
 - ブラウザからWebサーバ経由で、RESTやSOAPを利用する
- wgetコマンド
 - シェルなどからRESTを利用する
 - URLを指定して、結果をファイルに保存する

スクリプト言語でのREST,SOAPの利用

- スクリプト言語からのRESTの利用
 - HTTPさえ使えば、問題ない
 - HTTPが使えなくても、wgetコマンドを使って何とかなる

© XML Consortium

25

スクリプト言語でのREST,SOAPの利用

- スクリプト言語からのSOAPの利用
 - 型が明確ではない
 - WSDL2Javaのようなツールが存在しない
 - 開発時にコンパイラによるチェックができない
 - スキーマから自動生成されたクラスを使わなければ、送信するXMLが正しい事が保障できない
 - 必ず正しいリクエストが送信されるとは限らない
 - WS-*が使えるのか？
 - HTTP以外が使えるのか？
 - SOAP1.1以降、プロトコルはHTTPだけとは限定していない

↓

- SOAPの一部機能しか使えない

© XML Consortium

26



REST と SOAP

SOAPをREST対応にする方法



SOAPをREST対応にする方法



- Webサービス(SOAP)をコード追記無しでREST対応に
 - AXIS2
 - 自動的にRESTにも対応
 - <http://localhost:8080/axis2/services/version>
 - <http://localhost:8080/axis2/rest/version>
 - <http://ws.apache.org/axis2/1.1/rest-ws.html>
 - .NET Framework
 - 1.1以降では GET, POST がデフォルトで無効
 - web.config に設定を追記する
 - <http://support.microsoft.com/kb/819267/ja>

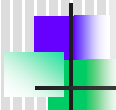
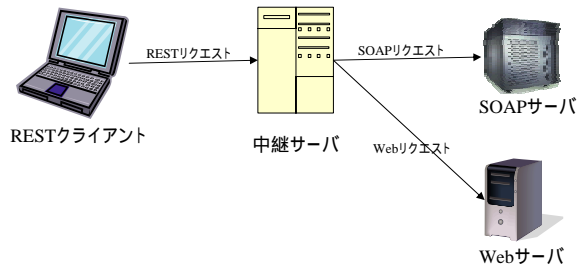
```
web.config
<configuration>
<system.web>
<webServices>
<protocols>
<add name="HttpGet"/>
<add name="HttpPost"/>
</protocols>
</webServices>
</system.web>
</configuration>
```



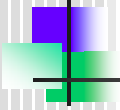
SOAPをREST対応にする方法



- RESTリクエストを中継してSOAPサーバに送る
 - 既存のWebサービス(SOAP)が変更不能な場合
 - 呼び出し先が非同期呼び出しの場合には、注意が必要
- 同様に、Webサーバに送った場合
 - 戻ってきたHTMLをXHTMLに変換する事でデータとして利用



REST と SOAP マッシュアップ

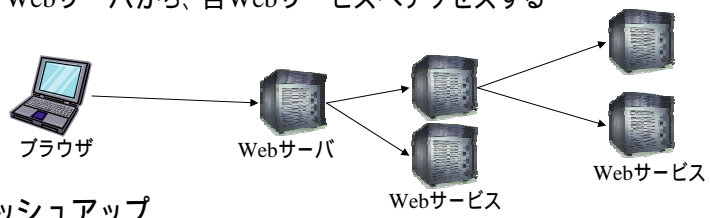


マッシュアップについて

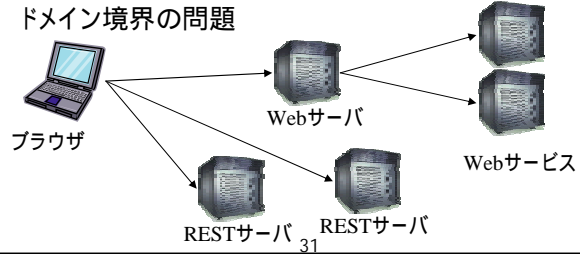


XML Consortium

- Webサービスでのアグリゲーション
 - Webサーバから、各Webサービスへアクセスする

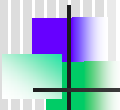


- マッシュアップ
 - ブラウザから直接アクセス、もしくは、Webサーバを経由する
 - ドメイン境界の問題



© XML Consortium

31



XML Consortium

今後の動向とまとめ

© XML Consortium

32

今後の動向



- Java
 - RESTとSOAPを併用した例
 - WSIT (Web Services Interoperability Technology)
 - <https://wsit.dev.java.net/>
 - JAX-WS2.0を利用した実装
 - 様々なWS-*に対応
 - SOAP呼び出しを行う直前に、GETでWSDLを取得している
 - JAXB2.0はスキーマに依存しなくてもいい
 - スクリプト言語のように、型の変化に柔軟に対応
- JavaScript
 - JSON (JavaScript Object Notation)
 - XMLではなく、JavaScriptが理解し易い形式で返す
 - ブラウザのJavaScriptでのDOM操作が不要に
- Perl
 - Perl5では、WSDLを読み込んでSOAPリクエストを自動生成する
 - 正しくない属性をセットしても、リクエスト時に送信されない
- スクリプト言語
 - 型を定義できるスクリプト言語 (Haskellなど) も現れた

まとめ



- REST
 - WADLにより、インターフェイスが明文化されるようになるのか？
 - 個人サイトで利用するには、手軽に使えるのが一番
- SOAP
 - WSDL1.1は普及しているが、いつ2.0になるのか？
 - WS-* の機能が必要であればSOAP
 - ミドルウェアによって、様々な仕様に対応していける
 - ミドルウェアが必要なので、手軽には使えない場合も
 - 高機能ゆえに、最初の学習コストが高い
- REST, SOAPとも、相手の長所を取り込み続けている