

SOAP

(この文書は EMBL-EBI の http://www.ebi.ac.uk/Tools/webservices/tutorials/03_soap の和訳です。このドキュメントは、12/09/14 時点の情報にもとづいて書かれています。このドキュメントは ecobioinfo.com で独自に訳したもので、EMBL-EBI とは直接関係ありません。)

Simple Object Access Protocol

SOAP (Simple Object Access Protocol) ^{1) 2)} は、HTTP (Hypertext Transfer Protocol) ^{3) 6)} あるいは、SMTP (Simple Mail Transfer Protocol) ⁷⁾ の上位層で、例外処理システム (faults) を含む、XML ^{3) 4)} ベースのメッセージングシステムを規定しています。

SOAP のリクエストとレスポンスは、同じ基本構造: SOAP エンベロープがオプションの SOAP ヘッダ、SOAP ボディ、オプションの SOAP エラー (SOAP Fault) を同封するという構造を持っています。HTTP をトランスポートとして用いる場合は、追加の HTTP ヘッダセット (例えば、SOAPAction) もあります。ボディ部は XML ドキュメントとしてエンコードされたメッセージのペイロードを含んでいます。多くの場合、どのように SOAP メッセージを構築、デコードするかの詳細を扱うソフトウェアライブラリが使われていますが、デバッグの時には、SOAP メッセージの構造にある程度は熟知していると役に立ちます。

SOAP スタイル

サーバとクライアントの間で交換される SOAP メッセージは、サービスの SOAP スタイルに依存する幾つかの異なる書式で生成することができます。

- Document/encoded: 使用されない
- Document/literal: メッセージは、XML スキーマに従い定義され書式化され、それ故、有効となる。
- RPC/encoded: メッセージは、オペレーションの名前を使った要素でラップされ、データは、XML スキーマとは独立してエンコードされ、型情報を含む。
- RPC/literal: メッセージは、オペレーションの名前を使った要素でラップされ、コンテンツは定義された XML スキーマに従うが、要素をラップするので直接的には有効にできないデータとなる。

SOAP ツールキットは、しばしスタイルのサブセットをサポートしているので、SOAP スタイルには、Web サービスをアクセスするときに使われるツールキットの選択の影響があります。The Web Services Interoperability Organization (WS-I) Basic Profile ⁸⁾ のような互換性と相互接続性のガイドラインは、しばし、RPC/encoded スタイルを危険性があるメッセージのエンコーディングの多様性として排除しています。非常に多くの、より新しい SOAP ツールキットが RPC/encoded のサポートを止め、そして、新しいサービスで Document/literal あるいは RPC/literal スタイルを使うことが一般的な動きとなっています。より古いサービスは、通常 RPC/encoded スタイルに基づいているので、このスタイルをサポートしている SOAP ツールキットを注意深く選択することが求められます。

SOAP スタイルのより詳細な情報は、[Which style of WSDL should I use?](#) を参照してください。

Document/literal のメッセージ

[EB-eye service](#) (訳注:EBI の情報資源に高速アクセスするサービス)は、document/literal のメッセージ(実際にはラップされた document/literal)を使います。例えば、listDomains()を使った操作の結果、交換されたメッセージ:

- リクエスト:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ebis="http://www.ebi.ac.uk/EBISearchService">
  <soapenv:Header/>
  <soapenv:Body>
    <ebis:listDomains/>
  </soapenv:Body>
</soapenv:Envelope>
```

- レスポンス:

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <ns1:listDomainsResponse
      xmlns:ns1="http://www.ebi.ac.uk/EBISearchService">
      <ns1:arrayOfDomainNames>
        <ns1:string>2can</ns1:string>
        <ns1:string>arrayexpress-experiments</ns1:string>
        <ns1:string>arrayexpress-genes</ns1:string>
        ...
        <ns1:string>uniref100</ns1:string>
        <ns1:string>uniref50</ns1:string>
        <ns1:string>uniref90</ns1:string>
      </ns1:arrayOfDomainNames>
    </ns1:listDomainsResponse>
  </soap:Body>
</soap:Envelope>
```

SOAP faults は、document/literal を使ってエンコーディングされます。例えば、その操作に無効なパラメータが渡されたことによって発生し、伝播した Java の例外:

```
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:Server</faultcode>
      <faultstring>Fault: java.lang.NullPointerException</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

これらのメッセージは、[soapUI](#)から引用したものです。

RPC/encoded のメッセージ

[WSDbfetch service](#) (訳注: accession No 等からエントリ情報を獲得するサービス)は、RPC/encoded メッセージを使っています。例えば、`getFormatStyles(db, format)`を使った操作の結果、交換されたメッセージ:

- リクエスト:

```
<soapenv:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:wsd="http://wsdbfetch.ws.jdbfetch.ebi.ac.uk">
  <soapenv:Header/>
  <soapenv:Body>
    <wsd:getFormatStyles
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <db xsi:type="xsd:string">uniprot</db>
      <format xsi:type="xsd:string">fasta</format>
    </wsd:getFormatStyles>
  </soapenv:Body>
</soapenv:Envelope>
```

- レスポンス:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getFormatStylesResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="http://wsdbfetch.ws.jdbfetch.ebi.ac.uk">
      <getFormatStylesReturn
        soapenc:arrayType="xsd:string[3]"
        xsi:type="soapenc:Array"
        xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
        <getFormatStylesReturn
          xsi:type="xsd:string">default</getFormatStylesReturn>
        <getFormatStylesReturn
          xsi:type="xsd:string">html</getFormatStylesReturn>
        <getFormatStylesReturn
          xsi:type="xsd:string">raw</getFormatStylesReturn>
        </getFormatStylesReturn>
      </ns1:getFormatStylesResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

SOAP fault の書式は、仕様の一部として定義されているので、常に `document/literal` でエンコードされます。例えば、`getFormatStyles(db, format)` の操作で、間違ったデータベース名を使った場合 [WSDbfetch service](#) は例外を上げます:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server.userException</faultcode>
      <faultstring>uk.ac.ebi.jdbfetch.exceptions.DbParamsException:
Database, uniprota , not found!</faultstring>
      <detail>
        <ns1:hostname xmlns:ns1="http://xml.apache.org/axis/">tomcat-
2.ebi.ac.uk</ns1:hostname>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

これらのメッセージは、[soapUI](#) から引用したものです。

SOAP サービスの解説

SOAP Web サービス API を記述するための現在存在する一連の一般的な仕組み:

- Really Simple Web Service Descriptions (RSWS) ⁹⁾
- SOAP Service Description Language (SSDL) ¹⁰⁾
- Web Services Description Language Version 1.1 (WSDL) ¹¹⁾
- Web Services Description Language Version 2.0 (WSDL 2.0) ¹²⁾

ほとんどの SOAP Web サービスには、WSDL に基づいたサービス記述文書がありますが、WSDL 2.0 では、例えば、RDF ¹³⁾ の SPARQL プロトコルとの SOAP バインディング仕様などの採択を考慮しています。

次のセクションでは WSDL 1.1 についてより詳細に見ていきます。

¹⁾ SOAP 1.1 - <http://www.w3.org/TR/soap11/>

²⁾ SOAP 1.2 - <http://www.w3.org/TR/soap12/>

³⁾ Extensible Markup Language (XML) 1.0 - <http://www.w3.org/TR/xml/>

⁴⁾ Extensible Markup Language (XML) 1.1 - <http://www.w3.org/TR/xml11/>

⁵⁾ RFC1945 - Hypertext Transfer Protocol – HTTP/1.0 - <http://www.faqs.org/rfcs/rfc1945.html>

⁶⁾ RFC2616 - Hypertext Transfer Protocol – HTTP/1.1 - <http://www.faqs.org/rfcs/rfc2616.html>

⁷⁾ RFC5321 - Simple Mail Transfer Protocol - <http://www.faqs.org/rfcs/rfc5321.html>

⁸⁾ WS-I Basic Profile 1.1 - <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>

⁹⁾ Really Simple Web Service Descriptions (RSWS) - <http://www.xml.com/lpt/a/1302>

¹⁰⁾ SOAP Service Description Language (SSDL) - <http://www.ssd1.org/>

¹¹⁾ Web Services Description Language Version 1.1 (WSDL) - <http://www.w3.org/TR/wsdl>

- [12\)](http://www.w3.org/TR/wsdl20/) Web Services Description Language Version 2.0 (WSDL 2) - <http://www.w3.org/TR/wsdl20/>
- [13\)](http://www.w3.org/TR/rdf-sparql-protocol/) SPARQL Protocol for RDF - <http://www.w3.org/TR/rdf-sparql-protocol/>
-