

# REST

(この文書は EMBL-EBI の [http://www.ebi.ac.uk/Tools/webservices/tutorials/02\\_rest](http://www.ebi.ac.uk/Tools/webservices/tutorials/02_rest) の和訳です。このドキュメントは、12/09/09 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](http://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。)

## Representational State Transfer

---

当初は Ron Fielding <sup>1)</sup> によって述べられた、サービスのためのアーキテクチャスタイルである REST (Representational State Transfer) は、HTTP (Hypertext Transfer Protocol) <sup>2) 3)</sup> の開発に端を発するものです。

REST の主な特徴は:

- クライアント-サーバ: クライアントがサーバベースのサービスのインターフェースとしての役割を果たす
- ステートレス: それぞれのリクエストは、リクエストを解釈するために必要な全ての情報を含んでいなければならない
- キャッシュ: レスポンスデータは、サーバ、クライアント、あるいは仲介するプロキシのキャッシング許可において、キャッシュ可能あるいはキャッシュ不可能に特徴付けられる
- 同一のインターフェース
  - リソースの識別: 個別のリソースにユニークな識別子 (例えば、URL や URI)
  - 表現を通してのリソースの操作: リクエストとレスポンスはオブジェクトの表現である (例えば、HTML や XML ドキュメント)
  - 自己表現メッセージ: メタデータの規定で表現のタイプを記述 (例えば、MIME タイプ)

これらの特徴は HTTP 設計の一部で、それ故 HTTP は REST プロトコルの一つであると考えられ、REST サービスは HTTP を基本としています。リソースの識別に URL を使うこと、データ表現型の識別に MIME タイプ、そして、リクエストの処理に必要な全ての状態を含むリクエストを使うことです。

## HTTP

---

現在、REST サービスは一般的に HTTP を使って実装されているので、HTTP プロトコルについてと、それがどのように働くか少しは理解しておくとい良いでしょう。

### 方法 (Method)

HTTP では、URL との情報交換に4種類のリクエスト方式が定義されています:

- GET: URL でアドレス指定されたリソースの読出し
- PUT: URL で特定されたアドレスに置くためのリソースを送信
- POST: URL にデータを送信して、状況に応じてレスポンスを受信
- DELETE: URL でアドレス指定されたリソースの削除

現在、REST Web サービスでは GET と POST しか使わない傾向があります。PUT と DELETE と違ってプロキシとファイアウォールで良くサポートされているからです。PUT と DELETE をサポートするサービスは、通常は互換性のために GET あるいは POST を通してそれらをサポートする代替のリソースが提供されています。

## ヘッダ領域

リクエストメソッドと URL には加えて、オプションのヘッダ値をリクエストに含めることができます。それらは、キー／値のペアとして実装され、リクエストとレスポンスの両方に載せることができます。

REST リクエストで重要と思われるヘッダ：

キー	解説
Accept	クライアントで受け付ける MIME タイプ
Content-Length	ボディコンテンツのサイズをバイトで表す
Content-Type	ボディコンテンツの MIME タイプ

一般的に使われるヘッダの一覧は [Wikipedia](#) にあります。

## ステータスコード

HTTP サーバからのレスポンスは、リクエストが成功したか、サービスが移動したか、あるいはエラーが起こったかを示すステータスコードを含んでいます。ステータスコードには、ユーザに見せるための簡単な解説も付いています。

一般的な例として含まれるもの：

コード	解説
200	OK (正常)
301	Moved Permanently (恒久的な移動)
307	Temporary Redirect (一時的なリダイレクト)
401	Unauthorized (認証されない)
404	Not Found (見つからない)
500	Internal Server Error (サーバ側のエラー)

標準ステータスコードの完全な一覧は [Wikipedia](#) で見つけられます。まれなケースとして、完全なプロトコルが Web サービスサーバ(例えば、DAS)ごとに実装されることがあり、それは標準コードが若干異なる意味を持つことを意味しています。その Web サービスで使われているステータスコードの詳細はその Web サービスのドキュメントで確認してください。

(訳注1:この場合の DAS は、direct-attached storage ではなく、Distributed Annotation System <sup>13)</sup> だと思われます。)

## データ表現

サービスから戻るデータは様々なフォーマットが可能です。返信されたデータをどのように扱うかのヒントは、コンテンツタイプヘッダの中から見つけ出せます。もし、ボディコンテンツの MIME タイプ <sup>4)</sup> ([Wikipedia](#)) を含むコンテンツタイプヘッダが定義されていたなら、それは適切なビューアある

はソフトウェアモジュールヘデータを受け渡すために利用できます。  
 幾つかのよく使われる MIME タイプ:

MIME	解説	拡張子
application/json	JSON( JavaScript Object Notation ) <sup>5)</sup> の文書	.json
application/octet-stream	バイナリーデータ	
application/xml	XML ( Extensible Markup Language ) <sup>6)</sup> 文書	.xml
chemical/x-pdb	PDB (タンパク質データバンク) <sup>7)</sup> 構造	.pdb
image/jpeg	JPEG <sup>8)</sup> イメージ	.jpg or .jpeg
image/png	PNG ( Portable Network Graphics ) <sup>9)</sup> イメージ	.png
text/html	XHTML ( Extensible HyperText Markup Language ) <sup>10)</sup> あるいは、HTML ( HyperText Markup Language ) <sup>11)</sup> 文書	.htm or .html
text/plain	普通のテキスト文書	.txt
text/xml	XML ( Extensible Markup Language ) <sup>12)</sup> 文書	.xml

MIME タイプの完璧なリストは、[IANA](#) から利用可能です。幾つかのアプリケーションは、外部アプリケーションによって操作されるデータを利用するため、非公式な MIME タイプを使うようです。

## エラー処理

どのようにクライアントとエラー通信するかは、REST サービスでは大変多様性のある領域の一つです。

1. 直接的に HTTP プロトコルを実装している Web サービスサーバ(例えば、DAS)は、エラー状態の交信で HTTP ステータスコードメカニズムを使用しています。
2. 幾つかのサービスは、エラー状態を知らせるための付加的な HTTP ヘッダを使用しています。
3. それ以外は、エラードキュメントをボディコンテンツの一部として使用しています(例えば、[dbfetch](#))。

使われている正確なメカニズムについては、サービスに付随するドキュメントに記述されているはずです。

## REST API ドキュメント

REST により柔軟に提案されたため、個々のサービスは独自の振る舞いとリソースの命名法を持っています。しかしながら、良く仕様が定義されていて広く実装されているいくつかのタイプのサービスがあります:

- AtomPub (<http://www.atomenabled.org/>): 分類されたデータと公開データのクエリ実行の API。AtomPub は、一般にはブログソフトウェアで使われていて、[Google Data API](#) のような API の基礎としても使われている。

- Distributed Annotation System (DAS) <sup>13)</sup> (<http://www.biodas.org/>): シーケンスデータと関連するフィーチャ・アノテーションを獲得するために使用する。
- OpenSearch (<http://www.opensearch.org/>): 検索を実行して結果を獲得するための API。OpenSearch は、一般には、Web ブラウザの検索プラグインに使われていて、検索結果の記述に RSS あるいは Atom フィードを使うかどうかの API としても使える。
- Search/Retrieval via URL (SRU) (<http://www.loc.gov/standards/sru/>): 検索実行と結果の獲得のための API。

これらの標準規格は、サービスの可能性を見出す仕組みと、実行時に呼び出す方法を提供しています。これは、クライアントを構築するときに、サービスとの通信にかかわる仕事の多くは、コード生成、ダイナミッククラスあるいはクライアントライブラリにより処理可能なので便利です。しかしながら、その仕組みでアクセスされているどのタイプのサービスが支援のために提供されているか知っておく必要があります。

REST 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:

- Norm's Service Description Language (NSDL) <sup>14)</sup>
- RESTful Services Description Language (Resedel) <sup>15)</sup>
- Simple Message Exchange Descriptor (SMEX-D) <sup>16)</sup>
- Web Application Description Language (WADL) <sup>17)</sup>
- Web Description Language (WDL) <sup>18)</sup>
- Web Resource Description Language (WRDL) <sup>19)</sup>
- Web Services Description Language Version 2.0 (WSDL 2.0) <sup>20)</sup>

最近 WADL では、[RESTDescribe/RESTCompile](#) のような幾つかの開発ツールの採択、JAX-RS <sup>21)</sup> の参考実装である [Jersey](#) での WADL の使用、そして、WADL ディスクリプションを提供している幾つかのサービスの登場を配慮しています。例えば:

- [WSDbfetch \(REST\)](#)
- [QuickGO](#)

WSDL 2.0 では、例えば: the SPARQL Protocol for RDF <sup>22)</sup> 仕様は WSDL 2.0 をインターフェースの定義に使うような利用も考慮しています。WSDL 2.0 を操作するツールが現れ、そして、Web サービスのツールキット(例えば、Apache Axis2 <sup>23)</sup>) が WSDL 2.0 のサポートを始めています。

<sup>1)</sup> [http://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)

<sup>2)</sup> RFC1945 - Hypertext Transfer Protocol – HTTP/1.0 - <http://www.faqs.org/rfcs/rfc1945.html>

<sup>3)</sup> RFC2616 - Hypertext Transfer Protocol – HTTP/1.1 - <http://www.faqs.org/rfcs/rfc2616.html>

<sup>4)</sup> RFC2046 - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types - <http://www.faqs.org/rfcs/rfc2046.html>

<sup>5)</sup> JavaScript Object Notation (JSON) - <http://www.json.org/>

<sup>6)</sup> , <sup>12)</sup> Extensible Markup Language (XML) - <http://www.w3.org/XML/>

<sup>7)</sup> Protein Data Bank (PDB) - <http://www ww p d b . o r g />

<sup>8)</sup> JPEG - <http://www.jpeg.org/jpeg/index.html>

<sup>9)</sup> Portable Network Graphics (PNG) - <http://www.w3.org/TR/PNG/>

<sup>10)</sup> Extensible HyperText Markup Language (XHTML) - <http://www.w3.org/TR/html/>

<sup>11)</sup> HyperText Markup Language (HTML) - <http://www.w3.org/TR/html4/>

<sup>13)</sup> Dowell R.D., Jokerst R.M., Day A., Eddy S.R. and Stein L. (2001)

### **The distributed annotation system.**

BMC Bioinformatics 2:7; DOI: [10.1186/1471-2105-2-7](https://doi.org/10.1186/1471-2105-2-7); PubMed: [11667947](https://pubmed.ncbi.nlm.nih.gov/11667947/)

<sup>14)</sup> Norm's Service Description Language (NSDL) - <http://norman.walsh.name/2005/03/12/nsdl>

<sup>15)</sup> RESTful Services Description Language (Resedel) -  
<http://recycledknowledge.blogspot.com/2005/05/resedel.html>

<sup>16)</sup> Simple Message Exchange Descriptor (SMEX-D) -  
<http://www.tbray.org/ongoing/When/200x/2005/05/03/SMEX-D>

<sup>17)</sup> Web Application Description Language (WADL) - <https://wadl.dev.java.net/>

<sup>18)</sup> Web Description Language (WDL) - <http://www.pacificspirit.com/Authoring/WDL>

<sup>19)</sup> Web Resource Description Language (WRDL) - <http://www.prescod.net/rest/wrdl/wrdl.html>

<sup>20)</sup> Web Services Description Language Version 2.0 (WSDL 2) - <http://www.w3.org/TR/wsdl20/>

<sup>21)</sup> JAX-RS - <https://jsr311.dev.java.net/>

<sup>22)</sup> SPARQL Protocol for RDF - <http://www.w3.org/TR/rdf-sparql-protocol/>

<sup>23)</sup> Apache Axis2 - <http://ws.apache.org/axis2/>

---