

# Introduction to Web Services

## (Web サービスの手引き)

(この文書は EMBL-EBI の [http://www.ebi.ac.uk/Tools/webservices/tutorials/01\\_intro](http://www.ebi.ac.uk/Tools/webservices/tutorials/01_intro) の和訳です。このドキュメントは、12/09/03 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。)

### Web サービスとは?

---

Web サービスという言葉は、いくつかの異なる意味と持ちます。(訳注 1)

1. 最も広い意義では、Web サービスとは Web から利用可能なサービスの全てです。
2. より一般的な意味では、Web サービスとは人間よりもむしろコンピュータプログラムから使われることを目的とした、Web 技術に基づいているサービスです。
3. 一部の例では、Web サービスとは特に、特定の Web サービス技術 (例えば、SOAP あるいは REST) を使った Web サービスを言います。

この手引き書全体としては、Web サービスという言葉はプログラムによって使われることを目的としたサービスを意味します。

Web 上では多くのバイオインフォマテックスのリソース(情報資源)が利用可能とはいえ、それらは通常はユーザが直接操作する Web インターフェースを通してしか利用できません。このことは、必要なデータやソフトウェアを地場の資材(訳注 2)にインストールして保守できない場合にシステムティックに情報資源へアクセスする必要があるアプリケーションの実用性を制限することになります。あるいは、データ資源や分析ツールを Web ポータルやワークベンチアプリケーションとして統合する必要がある場合です。このような場合に Web サービスは情報資源をリモートアクセスする手段を提供します。これは、リソースのメンテナンスコストを現場で消費するのではなく、サービスプロバイダに委託し、リソースからデータや結果を導き出すようなアプリケーション開発と配置のコストを大幅に削減する利点があります。

訳注 1. Web サービスの意味 2. 3. は Web API とも呼ばれます。

訳注 2. 現場の資材 “the local resources” は通常はパソコンであることが多いですが、

Web ページをユーザが直接操作できないバイオ機器や、その他の情報端末の場合もあります。

### クライアント/サーバ モデル

---

Web サービスは古典的なクライアントサーバモデルを使います:

サーバ(おそらく遠隔の)が、ユーザインターフェースを持つクライアントによって要求・利用されるリソースを提供します。簡単な例は Web を閲覧する場合で、Web ブラウザクライアントがユーザにページを表示し、ユーザがページを操作することを許可し、クライアントから要求があったときは Web サーバがページのデータ (HTML, images, etc) を提供します。

## 同期アクセスと非同期アクセス

多くのリソースはほぼ即時に結果を返すので、クライアントがリクエストを生成してサーバから結果を含むレスポンスが送信されるのを待つ同期的要求に適していますが、いくつかの分析には時間がかかります。そのような場合は、同期的要求はタイムアウトが発生してクライアントが無反応になるでしょう。この問題への対応として、そのようなリソースへのアクセスを供給するサービスは非同期的要求を構築するためのメカニズムを提供します。これらは通常はメソッドのコレクションの形でクライアントに対して、

1. ジョブの申請と、ジョブ識別子の獲得
2. ジョブの状態の獲得。一般的に、ジョブが中断・実行・終了・エラーかどうかを示すステータスコードの変換
3. 終了したジョブの結果を獲得

を許可します。

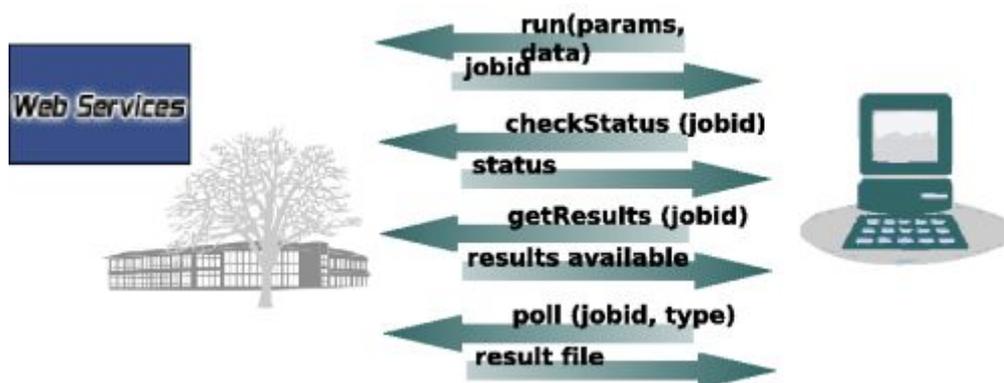
それらのメソッドとジョブのステータスコードの命名はサービスに依存し、ジョブのライフサイクルをより高度にコントロールする付加的なメソッドも提供されてる場合もあります。

### ディスパッチャ

EMBL-EBIでのディスパッチャを基本としたサービス(例えば、[WSFasta](#), [WSInterProScan](#), [WSNCBIBlast](#), [WSWUBlast](#))は、ジョブとの交信のための4つのメソッドを提供しています:

1. ジョブを申請するための "run" メソッド、これはサービス名で命名されます (例えば、runFasta for WSFasta)
2. ジョブの状態を獲得 (checkStatus)。ジョブの状態を示すステータスコードを返信します。見込まれる値:
  - RUNNING: ジョブは現在処理中
  - PENDING: ジョブは処理待ち行列の中
  - NOT\_FOUND: ジョブが見つからない, 通常はジョブがもはや利用可能でない場合にみられる
  - ERROR: ジョブが失敗
  - DONE: ジョブ終了, 結果を獲得することが可能
3. 終了したジョブの利用可能な結果タイプを検出 (getResults)
4. ジョブの指定した型の結果を獲得 (poll)

概要ワークフローは以下に示す通りです。



## Jdispatcher

EMBL-EBI の JDispatcher <sup>1)</sup>を基本としたサービス (例えば、[FASTA \(REST\)](#), [FASTA \(SOAP\)](#), [NCBI BLAST \(REST\)](#), [NCBI BLAST \(SOAP\)](#)) は、ジョブとの交信のための4つのメソッドを提供しています:

1. ジョブの申請 (run)
2. ジョブの状態を獲得 (getStatus)。ジョブの状態を示すステータスコードを返信します。  
見込まれる値:
  - RUNNING: ジョブは現在処理中
  - FINISHED: ジョブは終了, 結果を獲得することが可能
  - ERROR: ジョブステータスの獲得でエラーが発生。ほとんどのケースでは、ジョブは問題なく、何らかの問題でステータスへのアクセスが妨げられている、通常の回避方法は5秒待ち再実行
  - FAILURE: ジョブは失敗, 追加のエラー詳細がおそらく利用可能な出力にある
  - NOT\_FOUND: ジョブが見つからない, 通常の原因は誤ったジョブ識別子、あるいは、ジョブの有効期限切れで結果は消去されている
3. 終了したジョブの利用可能な結果タイプを検出 (getResultTypes)
4. ジョブの指定した型の結果を獲得 (getResult)

## Soaplab

[Soaplab](#) <sup>2)</sup> サービスは、ジョブとの交信のための多くのメソッドを提供しており、ジョブ非同期実行のためには3つの関連するメソッドがあります:

1. ジョブの申請 (一般のサービスには createAndRun、または、定型サービスには run)
2. ジョブの状態を獲得 (getStatus)。ジョブの状態を示すステータスコードを返信します。  
見込まれる値:
  - CREATED: ジョブは作成されたがまだ実行されていない
  - RUNNING: ジョブは現在処理中
  - COMPLETED: ジョブは終了, 結果を獲得することが可能
  - TERMINATED\_BY\_ERROR: ジョブがエラーにより終了した
  - TERMINATED\_BY\_REQUEST: ジョブがユーザまたは Soaplab ジョブマネージャにより終了させられた
3. ジョブの全ての結果を獲得 (getResults)

## Web サービス技術

---

コンピュータネットワークのリモートサービスにアクセスするための技術はたくさんあります。例えば:

- Common Object Request Broker Architecture (CORBA) <sup>3)</sup> ([Wikipedia](#))
- Distributed Computing Environment/Remote Procedure Call (DCE/RPC) ([Wikipedia](#))
- Distributed Ruby (DRb) <sup>4)</sup> ([Wikipedia](#))
- Extensible Messaging and Presence Protocol (XMPP) <sup>5)</sup> ([Wikipedia](#))
- Inter-Language Unification (ILU) <sup>6)</sup> ([Wikipedia](#))
- Internet Communications Engine (ICE) <sup>7)</sup> ([Wikipedia](#))

- Java Remote Method Invocation (Java RMI) ([Wikipedia](#))
- Jini <sup>8)</sup> ([Wikipedia](#))
- JSON-RPC <sup>9)</sup> ([Wikipedia](#))
- Open Network Computing Remote Procedure Call (ONC RPC or Sun RPC) ([Wikipedia](#))
- SOAPjr <sup>10)</sup> ([Wikipedia](#))
- SPARQL Protocol for RDF <sup>11)</sup> ([Wikipedia](#))
- XML-RPC <sup>12)</sup> ([Wikipedia](#))

これらの幾つかは、特定プログラム言語用(例えば、DRb や Java RMI)、オペレーティングシステム依存(例えば、DCE/RPC や ONC RPC)、あるいは、ネットワークのプロキシとファイアーウォールの問題があるような特別なネットワークプロトコル(例えば、CORBA や ILU)を利用しています。今日では、Web サービスで最も一般的に使われる技術は Representational State Transfer (REST) と Simple Object Access Protocol (SOAP) で、これらはどちらもプログラミング言語とオペレーティングシステムに非依存で、プロキシとファイアーウォールがよくサポートしている一般的なネットワークプロトコルを使います。

<sup>1)</sup> Goujon, M., McWilliam, H., Li, W., Valentin, F., Squizzato, S., Paern, J. and Lopez, R. (2010) **A new bioinformatics analysis tools framework at EMBL-EBI** Nucleic Acids Research; DOI: [10.1093/nar/gkq313](https://doi.org/10.1093/nar/gkq313); PubMed: [20439314](https://pubmed.ncbi.nlm.nih.gov/20439314/)

<sup>2)</sup> Senger M., Rice P., Bleasby A. and Uludag M. (2009) **Soaplab: Open Source Web Services Framework for Bioinformatics Programs** The 10th Annual Bioinformatics Open Source Conference; [PDF](#)

<sup>3)</sup> Common Object Request Broker Architecture (CORBA) - <http://www.corba.org/>

<sup>4)</sup> Distributed Ruby (DRb) - <http://segment7.net/projects/ruby/drbb/>

<sup>5)</sup> Extensible Messaging and Presence Protocol (XMPP) - <http://xmpp.org/>

<sup>6)</sup> Inter-Language Unification (ILU) - <ftp://ftp.parc.xerox.com/pub/ilu/ilu.html>

<sup>7)</sup> Internet Communications Engine (ICE) - <http://zeroc.com/ice.html>

<sup>8)</sup> Jini - <http://www.jini.org/>

<sup>9)</sup> JSON-RPC - <http://json-rpc.org/>

<sup>10)</sup> SOAPjr - <http://soapjr.org/>

<sup>11)</sup> SPARQL Protocol for RDF - <http://www.w3.org/TR/rdf-sparql-protocol/>

<sup>12)</sup> XML-RPC - <http://www.xmlrpc.com/>