

Apache Axis

(この文書は EMBL-EBI の

http://www.ebi.ac.uk/Tools/webservices/tutorials/06_programming/java/soap/apache_axis の和訳です。このドキュメントは、12/09/27 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。)

はじめに

Apache Axis ¹⁾ は、W3C ²⁾ に提出された SOAP ²⁾ の実装で、Apache SOAP プロジェクトの後続です。Apache Axis は JAX-RPC 標準規格 ⁴⁾ の実装です。

Apache Axis の一つの重要な特徴は、RPC/encoded と Document/literal スタイルの両方の SOAP サービスをサポートしていること、そして、両方のスタイルをサポートした WSDL to Java コードジェネレータを含んでいることです。

ご注意:

Apache Axis2 ⁵⁾ はここで説明する Apache Axis のバージョンではありません。

(訳注: Axis2 は Axis を一から書き直したものだそうです。)

インストール

Apache Axis 1.x は、<http://ws.apache.org/axis/> からダウンロードできます。

Apache Ivy

Apache Ivy ⁶⁾ を使う場合は、下記の依存宣言が使えます:

```
<!-- Apache Axis 1.x -->
<dependency org="commons-logging" name="commons-logging" rev="1.1.1" />
<dependency org="wsdl4j" name="wsdl4j" rev="1.6.2" />
<dependency org="commons-discovery" name="commons-discovery" rev="0.2" />
<dependency org="org.apache.axis" name="axis" rev="1.4" />
<dependency org="org.apache.axis" name="axis-jaxrpc" rev="1.4" />
```

Apache Maven

Apache Maven ⁷⁾ を使う場合は、下記の依存宣言が使えます:

```
<dependency>
  <groupId>commons-logging</groupId>
  <artifactId>commons-logging</artifactId>
  <version>1.1.1</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>axis</groupId>
  <artifactId>axis</artifactId>
  <version>1.4</version>
  <scope>compile</scope>
```

```
</dependency>
<dependency>
  <groupId>org.apache.axis</groupId>
  <artifactId>axis-jaxrpc</artifactId>
  <version>1.4</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>org.apache.axis</groupId>
  <artifactId>axis-saaj</artifactId>
  <version>1.4</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>axis</groupId>
  <artifactId>axis-wsdl4j</artifactId>
  <version>1.5.1</version>
  <scope>compile</scope>
</dependency>
<dependency>
  <groupId>commons-discovery</groupId>
  <artifactId>commons-discovery</artifactId>
  <version>0.4</version>
  <scope>compile</scope>
</dependency>
```

EMBL-EBI サンプルクライアント依存関係

Apache Axis 1.x ベースのサンプルクライアントのために必要な全ての依存関係を含むアーカイブを提供します: [lib-1.4.zip](#)。

これには開発ツールが含まれていて、クライアントを実行するときに `-Djava.ext.dirs` オプションで使えます。例えば、仮に `lib-1.4.zip` アーカイブがカレントディレクトリに解凍 (`lib` サブディレクトリを作成)され、`WSDbfetch` サンプルクライアント `jar` がカレントディレクトリにあるとすると:

```
java -Djava.ext.dirs=lib -jar WSDbfetch_Axis1.jar
```

でクライアントを実行できます。

WSDL2Java を使ってスタブを作成

Axis WSDL2Java ツールは、WSDL インターフェース定義からサービスインターフェースのクライアントクラスを生成します。デフォルトでは生成された `.java` ファイルは、WSDL で使われている名前空間に基づいたパッケージへ配置されます。クラス生成で特別なパッケージを指定するためには `-p` オプションを使用してください。

[EBI Web Services](#) で利用可能な WSDL 文書の詳細は、興味のあるサービスのサービスパッケージを参照してください。

注: Axis WSDL2Java で生成されたスタブは、バージョンによって異なるものです。もしサンプルク

クライアントで異なるバージョンの Axis を使いたいなら、スタブを生成して再コンパイルをしなければなりません。

コマンドライン

WSDL からスタブを生成、[WSDbfetch](#) の場合:

```
java org.apache.axis.wsdl.WSDL2Java http://www.ebi.ac.uk/ws/services/WSDbfetch?wsdl
```

このケースでは、スタブは `uk.ac.ebi.www` パッケージ内に生成されます。

スタブを強制的に `uk.ac.ebi.webservices.wsdbfetch` パッケージに入れるためには:

```
java org.apache.axis.wsdl.WSDL2Java -p uk.ac.ebi.webservices.wsdbfetch
http://www.ebi.ac.uk/ws/services/WSDbfetch?wsdl
```

Apache ant: generic task

追加タスクのインストール無しで、Apache ant [8](#) ビルドツールを使ってスタブをビルドするために、WSDL 文書からスタブコードを生成する包括的なターゲットの定義ができます:

```
<!-- Generate Axis 1.x stubs from a WSDL -->
<target name="genAxis1Stubs" description="Axis 1.x: generate stubs from a WSDL using
WSDL2Java class">
  <echo message="${stubPkg}" />
  <echo message="${wsdlUrl}" />
  <java classname="org.apache.axis.wsdl.WSDL2Java" classpathref="classpath"
    fork="yes" dir="${src.dir}">
    <arg value="-p" />
    <arg value="${stubPkg}" />
    <arg value="${wsdlUrl}" />
  </java>
</target>
```

これは `org.apache.axis.wsdl.WSDL2Java` クラスがクラスパスで定義されていること仮定しています。

それから `genAxis1Stubs` ターゲットは、スタブ生成のためのサービス固有ターゲットでパラメータ呼出しが可能です。例えば [WSDbfetch](#) では:

```
<!-- Generate Axis 1.x stubs for WSDbfetch -->
<target name="axis1-stubs-wsdbfetch" description="Axis 1.x: generate stubs
for WSDbfetch">
  <antcall target="genAxis1Stubs">
    <param name="stubPkg" value="uk.ac.ebi.webservices.axis1.stubs.wsdbfetch" />
    <param name="wsdlUrl"
      value="http://www.ebi.ac.uk/ws/services/WSDbfetch?wsdl" />
  </antcall>
</target>
```

次に、スタブ生成のために `ant` を呼び出すときターゲットを参照させることができます。

```
ant axis1-stubs-wsdbfetch
```

この方法で Apache Axis 1.x 用にスタブを生成するときを使う Apache ant の `build.xml` ファイルは、練習プロジェクト [java_exercises.zip](#) に含まれています。この仕組みを使うことによって、プロジェクトは Apache Ivy configuration によって操作される最少セットの依存関係を持つことになります (ant 用の Ivy プラグインは、ビルド処理の間に動的にダウンロードされインストールに追加されます)。

練習1: サービススタブを生成

練習プロジェクトに含まれる Apache ant ビルドファイル (`build.xml`) を調べてください。多くの ant タスクが見られ、それらはグループ:

- 環境設定
- ビルドディレクトリの消去
- 依存関係アーカイブを獲得して解凍
- WSDL から Axis スタブを生成
- ソースコードのコンパイル

に分けられます。

Eclipse で ant ビルドファイルに記述されたタスクを実行するためには、`build.xml` ファイルを右クリック、“実行” → “Ant ビルド...” で必要とするターゲットを選択してください。

Apache Axis のサンプルをビルドするには 2 つのターゲットを実行する必要があります:

1. `axis1-stubs`: コードスタブの生成
2. `axis1-compile`: 生成されたスタブとクライアントコードのコンパイル

これらを Eclipse で実行するには:

1. `build.xml` ファイルを右クリックしてメニューを表示
2. 利用可能なターゲットを表示するダイアログを開くため、“実行” → “Ant ビルド...” を選択
3. スタブ生成のために `axis-stub` ターゲットを選択
4. “順序...” ボタンを使って、`axis1-stub` を “`compile`” の前へ
5. “実行” ボタンをクリックして ant ビルドを呼び出し
6. 生成されたコードを Eclipse に確実に認識させるためにプロジェクトをリフレッシュ:
 1. プロジェクト (例えば `EBIWS_Java_Exercises`) を右クリック
 2. “リフレッシュ” を選択

これらのターゲットをコマンドラインから実行するには:

```
ant axis1-stubs
ant axis1-compile
```

Apache ant: Axis-wsdl2java task

Apache ant では、プラグインメカニズムを通して固有のタスクを追加することが許されています。Axis-wsdl2java タスク (<http://ws.apache.org/axis/java/ant/axis-wsdl2java.html>) は、WSDL 文書からの Axis スタブの生成をサポート、そして、もしインストールされているなら、上の一般タスクと入れ替えて使うことができます。

Apache maven

Maven プラグインも WSDL 文書から Axis コードスタブを生成するために利用可能です。詳細は、<http://maven-plugins.sourceforge.net/maven-axis-plugin/index.html> を参照してください。

スタブの利用

Axis WSDL2Java によって生成されたスタブは、WSDL で記述され相当する Java 言語に変換されたインターフェースとデータタイプの直接的なマッピングです。生成されたコードに目を通すことと、[WSDbfetch service documentation](#) と <http://www.ebi.ac.uk/ws/services/WSDbfetch?wsdl> は以下のセクションで役立つでしょう。

プログラムに生成されたクラスをインポートしてください (uk.ac.ebi.webservices.wsdbfetch パッケージを想定):

```
import uk.ac.ebi.webservices.wsdbfetch.*;
```

サービスと交信するサービスプロキシを生成するためには:

```
WSDBFetchServerService service = new WSDBFetchServerServiceLocator();  
WSDBFetchServer srvProxy = service.getWSDBFetch();
```

サービスプロキシのメソッド (srvProxy) は WSDL で定義されたものとマップして、WSDBFetch サービスの fetchData メソッドを呼出して使用できます:

```
String result = srvProxy.fetchData("UNIPROT:ADH1A_HUMAN", "default", "raw");
```

fetchData メソッドは、要求されたデータベースエントリを含む文字列を返します。これは:

```
System.out.println(result);
```

での出力と同じになります。

そして通常の方法でコンパイルします:

1. Eclipse のような IDE の使用:
 - “自動的にビルド”が有効なら、クラスはソースファイルが保存されたときに

コンパイルされます。

- 別の方法としては、“プロジェクト”,”すべてビルド”を使ってプロジェクト内の全てのソースをコンパイルできます。

2. Apache ant と提供されている build.xml ファイルを使う:

```
ant axis1-compile
```

3. コマンドラインから Java コマンドを使う:

```
javac uk/ac/ebi/webservices/wsdbfetch/*.java
javac Dbfetch.java
```

そして動作確認のためプログラムを実行します:

1. Eclipse のような IDE を使う: ソースファイル (例えば Dbfetc.java) を右クリック、“実行”で“Java アプリケーション”を選択する。
2. コマンドラインから:
 - CLASSPATH に Apache Axis ライブラリが追加されている場合:

```
java Dbfetch
```

- lib (Eclipse プロジェクトかライブラリアーカイブで提供される) と呼ばれるライブラリディレクトリを使用している場合:

```
java -Djava.ext.dirs=lib Dbfetch
```

練習 2: WSDbfetch (SOAP)

WSDbfetch Web サービス (<http://www.ebi.ac.uk/Tools/webservices/services/dbfetch>) は、[dbfetch](#) と同等の基本的な機能性を提供していますが、SOAP インターフェースのコレクションを通していません。

サンプルの WSDbfetch クライアント (<examples/soap/axis1/Dbfetch.java>) を参考として利用して、アクセッション: P13569, P26361, P35071 で、fasta シーケンス書式で、[UniProtKB](#) エントリを獲得するサービスを使ってみてください。

解答例: <solutions/soap/axis1/Q4Dbfetch.java>

練習 3: WSDbfetch メタデータ

[WSDbfetch](#) サービスには多くのメタデータが含まれていて、利用できるデータベース名と、それぞれのデータベースで使われるフォーマットとスタイルのようなサービスについての情報を提供します (<http://www.ebi.ac.uk/Tools/webservices/services/dbfetch>)。

サンプルクライアント (<examples/soap/axis1/Dbfetch.java>) から始めて、[WSDbfetch](#) サービスの `getDbFormats()` メソッドを使って、[UniProtKB](#) データベースに利用可能なデータフォーマットは何

かを判断してください。

解答例: [solutions/soap/axis1/Q5Dbfetch.java](https://solutions.soap.axis1/Q5Dbfetch.java)

複雑なデータ構造

[WSDbfetch service](#) のメソッドは全て簡単な文字列のパラメータを使っています。多くのほかの EBI サービスは、より複雑な入力データ構造を使っています。例えば、[NCBI BLAST \(SOAP\)](#) では [run\(email, title, params\)](#) メソッドへ渡す多様なパラメータと入力シーケンスを含んだ構造が求められます:

```
// Populate input parameters structure
InputParameters params = new InputParameters();
params.setProgram("blastp");
String[] databaseList = { "uniprotkb_swissprot" };
params.setDatabase(databaseList);
params.setType("protein");
params.setSequence(">Seq1 Example search sequence\n"
    + "MKFLILLFNILCLFPVLAADNHGVPQGASGVDPITFDINSNQTGPAFLTAVEMAGV");
String email = "email@example.org"; // Your e-mail address
// Get the NCBI BLAST (SOAP) service proxy
JDispatcherService_Service service = new
    JDispatcherService_ServiceLocator();
JDispatcherService_PortType ncbiblast =
    service.getJDispatcherServiceHttpPort();
// Submit the job
String jobId = ncbiblast.run(email, "", params);
System.out.println("Job Id: " + jobId);
```

[run\(email, title, params\)](#) メソッドは、ジョブのステータス(例えば、RUNNING, FINISHED, ERROR)を獲得する [getStatus\(jobId\)](#) メソッド、終了したジョブの利用可能な結果の詳細を獲得する [getResultTypes\(jobId\)](#) メソッド、そして、ジョブの結果を獲得する [getResult\(jobId, type\)](#) で使えるジョブ識別子を返します。

```
// Poll until job has finished
String status = "RUNNING";
while (status.equals("RUNNING")) {
    Thread.sleep(3000); // Wait 3 seconds
    status = ncbiblast.getStatus(jobId); // check job status
    System.out.println(status);
}
// If the job completed successfully...
if (status.equals("FINISHED")) {
    // Get the text result
    byte[] resultbytes = ncbiblast.getResult(jobId, "out", null);
    String result = new String(resultbytes);
    // Output the result
    System.out.println(result);
}
```

練習 4:NCBI BLAST (SOAP)

今まで見てきた全てのサービスは比較的短い時間内に結果を返信することができました。結果が返信されるまでに極端な場合は何日もかかる分析ツールを実行するときには、このケースは当てはまりません。そのため、処理申請でその後にジョブの状態をチェックするため使うジョブ識別子を返信して、終了したら結果を獲得するように、これらのツールには非同期メソッドを使わなければなりません。シーケンスの相同性検索処理に適している NCBI BLAST を使った [NCBI BLAST \(SOAP\)](#) サービスの実行にはこのアプローチが要求されます。

サンプルクライアント ([examples/soap/axis1/NcbiBlastSoap.java](#)) をよく調べて、”run”, ”getStatus”, ”getResult” の処理の流れに注目してください。

サンプルクライアントを参考として利用して、UniProtKB の CFTR_MOUSE (uniprot:CFTR_MOUSE) を UniProtKB/SwissProt (uniprotkb_swissprot) に対して NCBI BLAST 検索 を実行してください。

解答例: [solutions/soap/axis1/Q8NcbiBlastSoap.java](#)

練習 5:NCBI BLAST (SOAP) メタデータ

[NCBI BLAST \(SOAP\)](#) サービスは、多くの種類のフォーマットで結果を返すことができます。[getResultTypes\(jobId\)](#) メソッドを使って、他の利用可能な結果のタイプを検出するようにクライアントを修正してください。

解答例: [solutions/soap/axis1/Q9NcbiBlastSoap.java](#)

プロキシ

一部の環境では、クライアントが外部のサービスに接続できるように HTTP プロキシを設定する必要があります。Apache Axis はプロキシ設定のための通常の Java メカニズムをサポートしています:

1. Java システムプロパティ

- JVM に設定

```
java -Dhttp.proxyHost=proxy.example.org -Dhttp.proxyPort=8080  
ExampleClientClass
```

- クライアントコードで設定

```
System.setProperty("http.proxyHost", "proxy.example.org");  
System.setProperty("http.proxyPort", "8080");
```

2. Java.net.Proxy と java.net.ProxySelector クラスを使用

詳細と実例は下記を参照:

- Networking Properties:
<http://download.oracle.com/javase/6/docs/technotes/guides/net/properties.html>
- Java Networking and Proxies:
<http://download.oracle.com/javase/6/docs/technotes/guides/net/proxies.html>
- Apache Axis network configuration:

User-Agent

HTTP クライアントは通常そのクライアントが何者かについての情報を提供し、もし必要なら特定のクライアントのみ異なる操作をするサービスを可能にし、サービスがどのように利用されているかについて、サービス提供者にいくらかの情報が与えられます。デフォルトで Apache Axis は HTTP User-Agent ヘッダ ([RFC2616 section 14.43](#) 参照) を Axis/1.4、バージョン番号 (1.4) は Apache Axis のバージョン、のように設定します。

User-Agent 文字列は Apache Axis JAR の中にある `org/apache/axis/1.4/resource.properties` ファイルに設定されます。

サービスエンドポイントと名前空間

Web サービスの代替インスタンスへのアクセスが必要とされた場合、サービスのエンドポイントアドレスを変更する必要があります。Apache Axis では必要なエンドポイントアドレスを指定してサービスプロキシのオブジェクトを獲得するとき、それらをオーバーライドできます。

```
String srvEndpoint = "http://wwwdev.ebi.ac.uk/ws/services/WSDbfetchDoclit";
WSDbFetchDoclitServerService service = new
    WSDbFetchDoclitServerServiceLocator();
WSDbFetchServer srvProxy = service.getWSDbfetchDoclit(
    new java.net.URL(srvEndpoint));
```

サンプルクライアント

[EMBL-EBI](#) の大部分の SOAP Web サービスには、サービスへのコマンドラインでのアクセスと、サンプルコードを提供するサンプルクライアントがあります。Java 用に、いくつかのクライアントは Apache Axis 1.x をベースにしています。

Document/literal SOAP

Service	Sample client
EB-eye	Executable jar: EBeye_Axis1.jar ; Source code: EBeyeClient.java
InterProScan (SOAP)	Executable jar: IPRScan_Axis1.jar ; Source: AbstractWsToolClient.java , IPRScanClient.java
NCBI BLAST (SOAP)	Executable jar: NCBIblast_Axis1.jar ; Source: AbstractWsToolClient.java , NCBIblastClient.java
WSDbfetch (SOAP)	Executable jar: WSDbfetch_Axis1.jar ; Source: WSDbfetchClient.java

RPC/encoded SOAP

Service	Sample client
WSDbfetch	Executable jar: WSDbfetch.jar ; Source: WSDbfetchClient.java

-
- 1) Apache Axis - <http://ws.apache.org/axis/>
 - 2) SOAP - <http://www.w3.org/TR/soap/>
 - 3) W3C - <http://www.w3.org/>
 - 4) JAX-RPC - <http://jcp.org/en/jsr/detail?id=101>
 - 5) Apache Axis2/Java - <http://axis.apache.org/axis2/java/core/>
 - 6) Apache Ivy - <http://ant.apache.org/ivy/>
 - 7) Apache Maven - <http://maven.apache.org/>
 - 8) Apache ant - <http://ant.apache.org/>
-