

## REST ライブラリ比較

JAX-RS の実装にはライブラリが必要。

Jersey や Restlet などいくつかライブラリがある。

各ライブラリの比較は以下の URL を参照。

<http://www.ibm.com/developerworks/jp/web/library/wa-apachewink3/>

## クライアント

<http://java6.blog117.fc2.com/blog-entry-72.html>

### Jersey を使う場合

TestClient.java

### java.net を使う場合

### Apache Commons の HttpClient ライブラリを使う場合

httpcomponents-client-4.5.1 使用

## サーバ

### Web サーバをプログラムで立てる場合

<http://qiita.com/noobar/items/a96e07e441241b1e0215>

<http://unageanu.hatenablog.com/entry/20090723/1248353703>

### ライブラリを準備する

JAX-RS を実装するためのライブラリはいくつかあるが、ここでは Jersey を使う。

### ダウンロード

<https://jax-rs-spec.java.net/>

jersey 以外にも JDK http server が必要。

### サービスクラスを書く

TestService.java

### Web サーバを起動するためのランチャを書く

Launcher.java

### Web サーバ実行確認

必要なクラスパスを通して

```
java test.Launcher
```

### ブラウザで

```
http://localhost:8080/test
```

で

```
Hello World!
```

と出たら OK。

## Web サーバを Tomcat で行う場合

<http://www.atmarkit.co.jp/ait/articles/1411/10/news105.html>

### tomcat の設定

tomcat を使えるようにしておく。

### ディレクトリ構成

WEB-INF

```
web.xml
classes
  test
    TestService.class
lib
  aopalliance-repackaged-2.4.0-b31.jar
  asm-debug-all-5.0.4.jar
  hk2-api-2.4.0-b31.jar
  hk2-locator-2.4.0-b31.jar
  hk2-utils-2.4.0-b31.jar
  javassist-3.18.1-GA.jar
  javax.annotation-api-1.2.jar
  javax.inject-2.4.0-b31.jar
  javax.servlet-api-3.0.1.jar
  javax.ws.rs-api-2.0.1.jar
  jaxb-api-2.2.7.jar
  jersey-client.jar
  jersey-common.jar
  jersey-container-servlet-core.jar
  jersey-container-servlet.jar
  jersey-guava-2.22.1.jar
  jersey-media-jaxb.jar
  jersey-server.jar
  org.osgi.core-4.2.0.jar
  osgi-resource-locator-1.0.1.jar
  persistence-api-1.0.jar
  validation-api-1.1.0.Final.jar
```

### WEB-INF/lib に jar ファイルをコピーする

WEB-INF/lib に jersey の api、 ext、 lib の jar をファイルをすべてコピーする。

### サービスクラスを書く

引数をもらう書き方と複数のサービス書く書き方

下の例では、add2 は

```
http://hostname/add
```

で、add メソッドを呼び

```
http://hostname/add/add2
```

で add2 メソッドを呼ぶ

## web.xml を書く

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2
_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3
_0.xsd"
  id="WebApp_ID" version="3.0">
  <display-name>JerseyRESTServer</display-name>
  <servlet>
    <servlet-name>Jersey REST Service</servlet-name>
    <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
    <init-param>
      <param-name>jersey.config.server.provider.packages</param-name>
      <!-- 自分で作成したパッケージを書く。複数ある場合は ; で区切る -->
      <param-value>test</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Jersey REST Service</servlet-name>
    <!-- /* でもいいし、 /hoge/* でもよい -->
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

## 実行確認

ブラウザで

<http://localhost:8080/rest/test>

URL の rest の部分は tomcat の設定次第。環境によって変わる。