

Pemanggilan RESTful *Web Service* Java dari Client PHP

Husni@trunojoyo.ac.id

Pendahuluan

RESTful web services termasuk aplikasi kelas ringan dan cepat aksesnya. Inilah alasan bagus mengapa REST API ini banyak digunakan untuk komunikasi dan pertukaran data antar pada aplikasi berbasis web. Adalah cukup mudah mengakses RESTful web service yang ditulis dalam bahasa atau framework Java dari dalam bahasa atau framework PHP. Tutorial ini akan menjelaskan cara pembuatan RESTful web service sederhana di Java dan kemudian cara mengaksesnya dari PHP. Aplikasi PHP bertindak sebagai client.

RESTful Web Service di Java

Ada server dan ada client. Pada kasus kita, server atau web service harus disiapkan lebih dahulu. Marikita bangun RESTful web service sederhana dengan Java.

Kebutuhan Software



Persyaratan utama untuk membuat RESTful web service menggunakan Java adalah:

Software:

Java ([Download](#)).

GlassFish server ([Download](#)).

Eclipse IDE ([Download](#)).

File-file Jar Jersey ([Download](#)), yaitu:

jersey-client.jar

jersey-common.jar

jersey-container-servlet.jar

jersey-container-servlet-core.jar

jersey-media-jaxb.jar

jersey-server.jar

servlet-api-2.4.jar

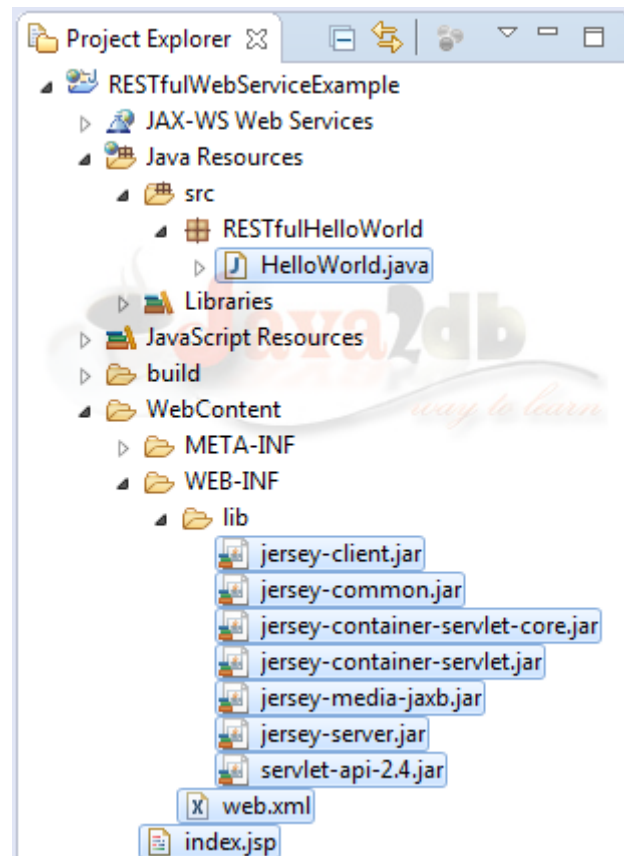


Dalam aplikasi contoh yang akan dibangun, kita menggunakan file-file jar **Jersey** yang dirilis oleh **Oracle**. Jika anda menginginkan, pustaka lain dapat digunakan seperti RESTEasy dari **Jboss** atau yang lainnya.

Langkah-langkah Pembuatan RESTful Web Service

► Langkah 1 :

Buat suatu proyek Dynamic Web di dalam IDE ([Eclipse](#) , [NetBeans](#) , [MyEclipse](#) atau [RAD](#)).
Buat suatu kelas Java ([HelloWorld.java](#)), file jsp ([index.jsp](#)) dan [web.xml](#) (lihat *source code*).
Tambahkan semua file jar ke dalam folder lib dari proyek kita. Silakan lihat struktur direktori di bawah!



► **HelloWorld.java**

```
1 package RESTfulHelloWorld;
2
3 import javax.ws.rs.GET;
4 import javax.ws.rs.Path;
5 import javax.ws.rs.Produces;
6 import javax.ws.rs.core.MediaType;
7
8 @Path("welcome")
9 public class HelloWorld {
10     @GET
11     @Produces(MediaType.TEXT_PLAIN)
12     public String getWelcomeNote() {
13         return "Congrats, you created first RESTful web service";
14     }
15 }
```

► web.xml

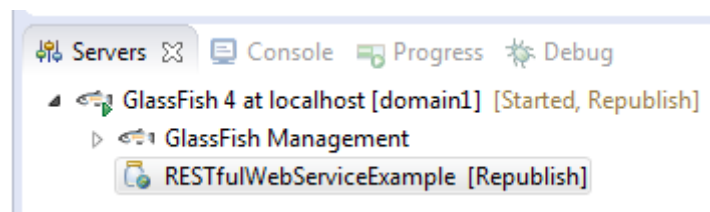
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- This web.xml file is not required when using Servlet 3.0 container,
3 see implementation details
4 http://jersey.java.net/nonav/documentation/latest/jax-rs.html -->
5 <web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
6 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
7 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
8 http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
9 <servlet>
10 <servlet-name>Jersey Web Application</servlet-name>
11 <servlet-class>org.glassfish.jersey.servlet.ServletContainer
12 </servlet-class>
13 <init-param>
14 <param-name>jersey.config.server.provider.packages</param-name>
15 <param-value>RESTfulHelloWorld</param-value>
16 </init-param>
17 <load-on-startup>1</load-on-startup>
18 </servlet>
19 <servlet-mapping>
20 <servlet-name>Jersey Web Application</servlet-name>
21 <url-pattern>/hello/*</url-pattern>
22 </servlet-mapping>
23 </web-app>
```

► index.jsp

```
1 <html>
2 <body>
3 <h2>Welcome to Hello World RESTful Web Application!</h2>
4 <p><a href="hello/welcome">Click Here</a>
5 </body>
6 </html>
```

► Langkah 2 :

Silakan deploy proyek web tersebut dalam suatu server web/aplikasi seperti [GlassFish](#), [Tomcat](#), [WebSphere](#), [Wildfly](#) atau [JBoss](#). Gunakan versi terbaru dan stabil.



► Langkah 3 :

Uji atau test aplikasi RESful web service dengan URL-nya melalui web browser.

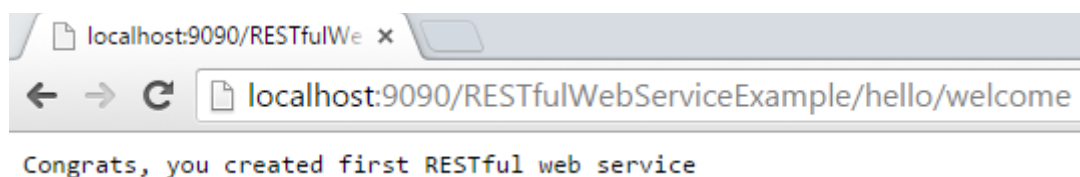
Masukkan URL (localhost:9090/RESTfulWebServiceExample/index.jsp) dalam address bar dari web browser.

Setelah address disubmit, akan muncul halaman berikut:



► **Langkah 4 :**

Klik link ([Click Here](#)) untuk memanggil sumber daya dari RESTful web service. Terakhir web service mengembalikan tampilan berikut:



Dari layar respon di atas, URInya adalah

- <http://localhost:9090/RESTfulWebServiceExample/hello/welcome>

Penjelasannya adalah

- **localhost** : nama atau alamat (IP address) dari web service: 127.0.0.1.
- **9090** : port dari server web.
- **RESTfulWebServiceExample** : nama proyek (nama aplikasi Web).
- **hello** : `<url-pattern>` yang dikonfigurasi di dalam file web.xml.
- **welcome** : path sumber daya yang dikonfigurasi di dalam kelas HelloWorld.java (`@Path("welcome")`).

Membuat Client RESTful dengan PHP

Sekarang saatnya membuat client RESTful API menggunakan PHP.

👉 **Langkah-langkah memanggil RESTful web service Java dari dalam client PHP adalah**

► **Langkah 1 :**

Buat suatu file PHP pada lokasi server (jika anda menggunakan XAMPP maka buat di bawah direktori htdocs).

Dalam contoh di berikut, kita buat di `C:\xampp\htdocs\phpServiceExe`. Filenya bernama `phpRESTfulClientForJavaService.php` (misalnya).

► **Langkah 2 :**

Salin kode berikut ke dalam file `phpRESTfulClientForJavaService.php`. File ini akan memanggil RESTful web service Java. Tidak diperlukan penambahan pustakan apapun untuk menjalankan aplikasi PHP ini.

```

1 <?php
2 //tidak perlu jar file
3 $url ="http://localhost:9090/RESTfulWebServiceExample/hello/welcome/";
4
5 // lewatkan URL dari RESTful Java-nya
6 $client = curl_init($url);
7 curl_setopt($client,CURLOPT_RETURNTRANSFER,1);
8
9 //curl_setopt($client,CURLOPT_POST,$data);
10 $response = curl_exec($client);
11
12 //$result = json_decode($response);
13 if(!empty($response)) {
14     echo $response;
15 } else {
16     echo "No data found";
17 }
18 ?>

```

► **Langkah 3:**

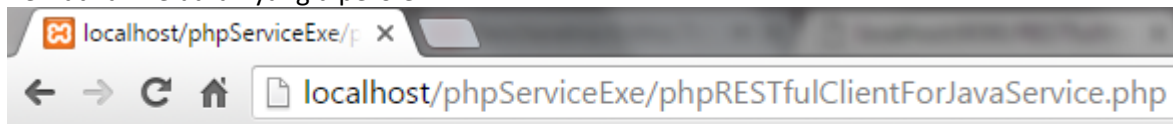
Jalankan web server (Apache) untuk menjalankan file phpRESTfulClientForJavaService.php.

► **Step 4 :**

Buka web browser dan masukkan URL dari file PHP tersebut.

- <http://localhost/phpServiceExe/phpRESTfulClientForJavaService.php>

Perhatikan keluaran yang diperoleh.



Congrats, you created first RESTful web service

Kita telah berhasil mengakses RESTful web service Java dari client yang ditulis dalam PHP.