Web-Based Service Optimization with JSON-RPC Platform in Java and PHP

WachyuHari Haji
Faculty of Computer Science, MercuBuana University
Jl. Meruya Selatan No. 1 Jakarta Barat, Indonesia
wahyuhari@gmail.com

Abstract—Web based application has been used by many organizations to support their business activities within the organization. Sometimes, that application requires the availability of information from other parties outside the organization. Other than that, there is a need to exchange information between applications in a particular organization with applications in other organizations. The applications may be on the different systems and platforms. There is a technology that can be used, commonly known as a web service. JSON-RPC is an alternative to build a web service. This technology involves two applications that are connected via a data connection. During the connection, an application can executes a remote procedure from web service provider to do implementation web service using JSON-RPC technology, with PHP platform for client side application and java platform for server side application.

Keywords: Web service, JSON-RPC, PHP, java

I. INTRODUCTION

At present the need for web-based applications become a major requirement for many organizations to support all business activities in the organization. Sometimes an application requires the availability of information from other parties outside the organization, then the information is used as data for further processing. In addition, there is a need to exchange information between applications in a particular organization with applications in other organizations, which are likely to be on different systems with different platforms. That is why we need a technology that enables cross-platform communication via the web. One technology that can be used is a web-based service technology or commonly known as a web service. This technology allows for cross-platform communication. Web-based services are used as a service provider in the form of information to other applications, so that between different applications can interact and communicate through the functions provided by an application service provider. This technology facilitates multiple applications to interact with other applications within an organization and outside organization using standards that are not tied to any platform and are not tied to the issue of what programming language used by each application, so that applications can access a sub-routine in Other applications, such as the

application is currently accessing the sub-routine in the local system.

RPC is a simple protocol that can be used to build a web-based services. RPC allows an application to call to the sub-routine provided by other applications residing on different systems. RPC uses a specific data exchange formats like XML or JSON as a communication liaison between systems. RPC protocol with JSON format as a data exchange format called JSON-RPC. This research will implement a web-based services with JSON-RPC for PHP platform using the Java platform client application and server application service provider.

II. WEB SERVICE

Web-based services is a technology that aims to facilitate communication between applications, which use a standard platform that is not bound and not bound by the programming language used by each application. W3C definition (2011) "A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in machine-processable format (specifically WSDL). Other systems use that interface description to create connections to the system. The system exposes services, which use a protocol specifically designed for that system (sometimes called a remoting protocol).".

a. JSON

JSON is a data exchange format which has the characteristics of lightweight, easily understandable by humans, and have translated easily by computer. JSON is based on the JavaScript programming language. JSON format itself is a text that is independent of the programming language of any kind, this is because the style of language used is a style that is commonly used by programmers for some programming languages. JSON includes C, C++, C#, Java, JavaScript, Perl, Python, etc. These properties make JSON ideal for use as a data exchange format. JSON uses the following form:

Object
A pair of name/value that is not Sorted. The object of writing begins with the character "{" (open brace) and ends with the character "}" (curly braces). Each name is always followed by each name/value pair separated by the character "," (comma).

**Array**

Arrays are a collection of values Sorted. Array always begins with the characters "[" (open square brackets) and ends with the characters "]" (close brackets). Each value is separated by the character "," (comma).

**Remote Procedure Call**

Remote Procedure Call (RPC) is a fairly simple protocol that is widely used today to build web-based services. According to Dave Marshall [1999], RPC defined "RPC is a powerful technique for constructing distributed, client-server based applications. It is based on extending the notion of convention al or local procedure calling. So that the called procedure need not exist in the same address space as the calling procedure. The two processes may be on the same system, or they may be on different systems with a network connecting them."

RPC allows us to access a sub-routine that is on another application within the application is located on the system and on different machines. To do this, an application server must provide a set of sub-routines which can be accessed by client applications. The approach is to facilitate communication over a socket, then the server will wait for a client that calls sub-routine provided by the server.

RPC protocol uses specific data format to exchange data, a data format that many use XML technology, the XML protocol is usually used with XML-RPC. However, over the technology, found an extension data exchange format JSON, which has advantages over XML. JSON, among others, the nature of light, which has advantages over XML. It is easy to read by a human being, as well as small size, so that the results will be more efficient than XML. It is certainly useful for later data processing, which will be with the fast er than XML. The RPC protocol which uses JSON as the data exchange format is called JSON-RPC.

d. **JSON-RPC**

The basic concept of JSON-RPC itself is very simple, which is a common mechanism consisting of two systems that are connected via data connection. During the connection, if the system is found, a system can perform an RPC routine and send a request to the server, then the server will give the object notification, the server will not give the object reply when the client sends a request. When an RPC is done, the input parameters mapped to sub-routine in the service provider server and to the client to call the sub-routine is waiting for another object that is returned from the sub-routine in the call.

**PHP**

PHP (Hypertext Preprocessor) is a web-based programming language that can process data dynamically. PHP will be fully executed by the server, which includes regular HTML pages. Applications are constructed by PHP in general, it will give the result to the web server, but the overall process is executed on the server. In principle, the server will work if there is an request from the client. In this case the client is using PHP, the code is executed. PHP, code is executed by the server. When using PHP, the scripting language of the application server, the server will do the following:

- Read request from client
- Search the site/page on the server
- Perform the instructions given by the PHP to make modifications to the page.
- Send back the pageto the client via the internet/intranet.

**Figure 1 JSON Object**

**Figure 2 JSON Array**

**Figure 3 Flow processes in JSON-RPC**
PHP has several advantages that can run on different platforms (Windows, Linux, Unix, etc.), but it's very easy to learn PHP.

f. **JAVASCRIPT**

JavaScript is a scripting language based on the concept of prototype-based programming. This language is particularly well known for its use in websites (as client-side JavaScript), and also used to provide access to objects embedded in another application.
III. GLOBAL WEB DESIGN BASED SERVICES

Making web-based services with JSON-RPC involves two applications that are connected via a data connection. During the connection, if the sub-routine exists, an application can make calls and the execution of the sub-routine is contained in another application. To make a call to a sub-routine in another system, the object of the client object will be sent by the client application to the application on the server, then the server will provide the results of processing the object notification against a client application based on the request object, unless the being sent by the client is the object of the notification, then the server will not send any reply to the client. Figure 4 below illustrates the flow chart for the process of communication that occurs between client and server on the system web-based services with JSON-RPC technology globally.

![Figure 4 Flowchart Web-Based Service Systems With JSON-RPC](image)

The implementation for this component is a Javascript file that will be placed in the client application. At this Javascript file there is a class that is JSONRpcClient class, which has a sub-b-routine specification as follow:

<table>
<thead>
<tr>
<th>No</th>
<th>Name of Sub-Routine</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>JSONRpcClient(vurl, vuser, vpassword)</td>
<td>Sub-routine is usefulearly initialization constructor and validation of URL if it is an application server.</td>
</tr>
<tr>
<td>2.</td>
<td>JSONRpcClient.prototype.doLogin = function()</td>
<td>Sub-routine is useful for logging onto the application server web-based service providers.</td>
</tr>
<tr>
<td>3.</td>
<td>JSONRpcClient.prototype.getDataFilm = function(id, namaBioskop)</td>
<td>Sub-routine is useful for showing films that are aired today at several theaters.</td>
</tr>
<tr>
<td>4.</td>
<td>JSONRpcClient.prototype.getDataBioskop = function(id)</td>
<td>Sub-routine is useful for displaying data existing cinema.</td>
</tr>
<tr>
<td>5.</td>
<td>function GetXmlHttpObject()</td>
<td>Sub-routine is useful for objects getting XMLHttpObject, which will be used for accessing web-based service providers server via AJAX technology.</td>
</tr>
<tr>
<td>6.</td>
<td>JSONRpcClient.prototype.getPooling = function(id)</td>
<td>Sub-routine is useful for displaying data polling.</td>
</tr>
<tr>
<td>7.</td>
<td>JSONRpcClient.prototype.setPooling = function(pilihan, id)</td>
<td>Sub-routine is useful for setting the selection of the most popular movie poll.</td>
</tr>
<tr>
<td>8.</td>
<td>function randomString()</td>
<td>Sub-routine is useful for getting a random string to be used as an ID on the request object.</td>
</tr>
</tbody>
</table>

### a. JSON-RPC SERVLET

Implementation of JSON-RPC Servlet component is in the form of a class named ServletJSONRPC which is a subclass of the Servlet class (javax.servlet.http.HttpServlet) in the java library. This class serves as a liaison between the client application to the application server. All sub-routines that exist in this class have the same as its parent class (javax.servlet.http.HttpServlet). with an sub-routine reimplementation of the sub-routine "doGet" and "doPost". Details can be seen in the following table:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**TABLE 1**

---

ISSN 2301-6590

1st International Conference on Engineering and Technology Development (ICETD 2012)
Universitas Bandar Lampung
Faculty of Engineering and Faculty of Computer Science
TABLE 2.
SUB-CLASS ROUTINESERVLETJSONRPC(JSON-RPC SERVLET)

<table>
<thead>
<tr>
<th>No</th>
<th>Name of Sub-Routine</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>doGet(HttpServletRequest req, HttpServletResponse resp)</td>
<td>Sub-routine will be called when an object request sent by the client through the delivery method &quot;GET&quot;</td>
</tr>
<tr>
<td>2.</td>
<td>doPost(HttpServletRequest req, HttpServletResponse resp)</td>
<td>Sub-routine will be called when an object request sent by the client through the delivery method &quot;POST&quot;</td>
</tr>
</tbody>
</table>

IV. TEST RESULTS

The results of direct access to the application server through the browser can be seen in the following table:

TABLE 3.
CLASS REQUEST HANDLER (OBJECT HANDLER FOR OBJECT REQUEST)

<table>
<thead>
<tr>
<th>No</th>
<th>Name of Sub-Routine</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>public String[] handledRequests()</td>
<td>Get names of all the sub-routine that handles all the request object</td>
</tr>
<tr>
<td>2.</td>
<td>public JSON RPC2 Response process(JSON RPC2 Request req)</td>
<td>Object to process incoming requests</td>
</tr>
</tbody>
</table>

TABLE 4.
NOTIFICATION HANDLER CLASS (OBJECT HANDLER FOR OBJECT NOTIFICATION)

<table>
<thead>
<tr>
<th>No</th>
<th>Method</th>
<th>Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>public String[] handledNotifications( )</td>
<td>Get names of all the sub-routine that handles all notification objects</td>
</tr>
<tr>
<td>2.</td>
<td>public process(JSON RPC2 Notification notification)</td>
<td>Processing notification of the entering Objects</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS

Based on the implementation of case studies have been conducted, it can be concluded as follows:

- The test result showed that the communication goes as expected, and the data is displayed on the server obtained from the server.
- The test result showed that the object is given by the server response was appropriate and also support the use of object return an error.

The results of direct access to the application server through the browser can be seen in the following table:

TABLE 5.
TEST RESULT THROUGH BROWSER APPLICATION SERVER

<table>
<thead>
<tr>
<th>Object request</th>
<th>Object replies</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;id&quot;:&quot;hDCV1dNX&quot;,&quot;method&quot;:&quot;getDat aBioskop&quot;,&quot;params&quot;:[],&quot;jsonrpc&quot;:&quot;2.0&quot;</td>
<td>{&quot;id&quot;:&quot;hDCV1dNX&quot;,&quot;result&quot;:{&quot;Citos&quot;,&quot;Grand Indonesia&quot;,&quot;Pejaten Village&quot;},&quot;jsonrpc&quot;:&quot;2.0&quot;}</td>
<td></td>
</tr>
</tbody>
</table>
REFERENCES


