

# XML-RPC - REQUEST FORMAT

[http://www.tutorialspoint.com/xml-rpc/xml\\_rpc\\_request.htm](http://www.tutorialspoint.com/xml-rpc/xml_rpc_request.htm)

Copyright © tutorialspoint.com

XML-RPC requests are a combination of XML content and HTTP headers. The XML content uses the data typing structure to pass parameters and contains additional information identifying which procedure is being called, while the HTTP headers provide a wrapper for passing the request over the Web.

Each request contains a single XML document, whose root element is a *methodCall* element. Each *methodCall* element contains a *methodName* element and a *params* element. The *methodName* element identifies the name of the procedure to be called, while the *params* element contains a list of parameters and their values. Each *params* element includes a list of *param* elements which in turn contain *value* elements.

For example, to pass a request to a method called *circleArea*, which takes a *Double* parameter *fortheradius*, the XML-RPC request would look like:

```
<?xml version="1.0"?>
<methodCall>
  <methodName>circleArea</methodName>
  <params>
    <param>
      <value><double>2.41</double></value>
    </param>
  </params>
</methodCall>
```

The HTTP headers for these requests will reflect the senders and the content. The basic template looks as follows:

```
POST /target HTTP 1.0
User-Agent: Identifier
Host: host.making.request
Content-Type: text/xml
Content-Length: length of request in bytes
```

For example, if the *circleArea* method was available from an XML-RPC server listening at */xmlrpc*, the request might look like:

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
```

Assembled, the entire request would look like:

```
POST /xmlrpc HTTP 1.0
User-Agent: myXMLRPCClient/1.0
Host: 192.168.124.2
Content-Type: text/xml
Content-Length: 169
<?xml version="1.0"?>
<methodCall>
  <methodName>circleArea</methodName>
  <params>
    <param>
      <value><double>2.41</double></value>
    </param>
  </params>
</methodCall>
```

It's an ordinary HTTP request with a carefully constructed payload.

