

# PHP - AJAX XML Parser

Using PHP with AJAX, we can parse an XML document from local directory as well as on a server. The following example demonstrates how to parse XML with web browser.

The client-end script renders a HTML form and defines a JavaScript function for sending a HTTP request to the server with XMLHttpRequest object.

On the server, a PHP script loads the DOM object from the required XML document, fetches the selected course from \$\_REQUEST variable, and renders the details of the course chosen as the response back to the client.

## Step 1

The following XML document is stored on the document root of the XAMPP server.

```
<?xml version = "1.0" encoding = "utf-8"?>
<CATALOG>

  <SUBJECT>
    <COURSE>Android</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$10</PRICE>
    <YEAR>2015</YEAR>
  </SUBJECT>

  <SUBJECT>
    <COURSE>Html</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$15</PRICE>
    <YEAR>2015</YEAR>
  </SUBJECT>

  <SUBJECT>
    <COURSE>Java</COURSE>
    <COUNTRY>India</COUNTRY>
    <COMPANY>TutorialsPoint</COMPANY>
    <PRICE>$20</PRICE>
```

```
<YEAR>2015</YEAR>
</SUBJECT>

<SUBJECT>
  <COURSE>Microsoft</COURSE>
  <COUNTRY>India</COUNTRY>
  <COMPANY>TutorialsPoint</COMPANY>
  <PRICE>$25</PRICE>
  <YEAR>2015</YEAR>
</SUBJECT>

</CATALOG>
```

## Step 2

The AJAX code below has a HTML form and a JavaScript function to raise HTTP request through XMLHttpRequest object.

&lt;/&gt;

Open Compiler

```
<html>
<head>
  <script>
    function showCD(str) {
      if (str == "") {
        document.getElementById("txtHint").innerHTML = "";
        return;
      }

      if (window.XMLHttpRequest) {

        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp = new XMLHttpRequest();

      } else {

        // code for IE6, IE5
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
      }

      xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
          document.getElementById("txtHint").innerHTML = xmlhttp.responseText
```

```

    }
}
xmlhttp.open("GET","hello.php?q="+str,true);
xmlhttp.send();
}
</script>
</head>
<body>
    <form>
        Select a Course:
        <select name = "cds" onchange = "showCD(this.value)">
            <option value = "">Select a course:</option>
            <option value = "Android">Android </option>
            <option value = "Html">HTML</option>
            <option value = "Java">Java</option>
            <option value = "Microsoft">MS technologies</option>
        </select>
    </form>
    <div id = "txtHint"><b>Course info will be listed here...</b></div>
</body>
</html>

```

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## Step 3

The server-side PHP script to search within the XML document is as follows –

```

<?php
    $q = $_GET["q"];

    $xmlDoc = new DOMDocument();
    $xmlDoc->load("test.xml");

    $x = $xmlDoc->getElementsByTagName('COURSE');

    for ($i = 0; $i<=$x->length-1; $i++) {
        if ($x->item($i)->nodeType == 1) {
            if ($x->item($i)->childNodes->item(0)->nodeValue == $q) {
                $y = ($x->item($i)->parentNode);
            }
        }
    }
}

```

```
    }  
  }  
}  
  
$cd = ($y->childNodes);  
  
for ($i = 0;$i<$cd->length;$i++) {  
  if ($cd->item($i)->nodeType == 1) {  
    echo("<b>" . $cd->item($i)->nodeName . ":</b> ");  
    echo($cd->item($i)->childNodes->item(0)->nodeValue);  
    echo("<br>");  
  }  
}  
?  
>
```

Visit "http://localhost/example.php" to let the user select a course. Upon selection, the relevant details are fetched from the server and displayed as below –

Select a Course:

**COURSE:** Java

**COUNTRY:** India

**COMPANY:** TutorialsPoint

**PRICE:** \$20

**YEAR:** 2015