

XML DOM - REPLACE NODE

http://www.tutorialspoint.com/dom/xml_dom_replace_node.htm

Copyright © tutorialspoint.com

In this chapter we will study about the replace node operation in an XML DOM object. As we know everything in the DOM is maintained in a hierarchical informational unit known as node and the replacing node provides another way to update these specified nodes or a text node.

Below are two methods to replace the nodes.

- `replaceChild`
- `replaceData`

replaceChild

The method *replaceChild* replaces the specified node with the new node.

Syntax

The `insertData` has following syntax:

```
Node replaceChild(Node newChild, Node oldChild) throws DOMException
```

Where:

- *newChild*: is the new node to put in the child list.
- *oldChild*: is the node being replaced in the list.
- This method returns the node replaced.

Example

The following example *replacenode_example.htm* parses an XML document ("[node.xml](#)") into an XML DOM object and replaces the specified node `<FirstName>` with the new node `<Name>`.

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function loadXMLDoc(filename)
      {
        if (window.XMLHttpRequest)
        {
          xhttp = new XMLHttpRequest();
        }
        else // code for IE5 and IE6
        {
          xhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xhttp.open("GET", filename, false);
        xhttp.send();
        return xhttp.responseXML;
      }
    </script>
  </head>
  <body>
    <script>
      xmlDoc=loadXMLDoc("/dom/node.xml");

      x = xmlDoc.documentElement;

      z=xmlDoc.getElementsByTagName("FirstName");
      document.write("<b>Content of FirstName element before replace operation</b><br>");
```

```

for (i=0;i<z.length;i++)
{
    document.write(z[i].childNodes[0].nodeValue);
    document.write("<br>");
}
//create a Employee element, FirstName element and a text node
newNode = xmlDoc.createElement("Employee");
newTitle = xmlDoc.createElement("Name");
newText = xmlDoc.createTextNode("MS Dhoni");

//add the text node to the title node,
newTitle.appendChild(newText);
//add the title node to the book node
newNode.appendChild(newTitle);

y = xmlDoc.getElementsByTagName("Employee")[0]
//replace the first book node with the new node
x.replaceChild(newNode,y);

z = xmlDoc.getElementsByTagName("FirstName");
document.write("<b>Content of FirstName element after replace operation</b><br>");
for (i = 0;i<z.length;i++)
{
    document.write(z[i].childNodes[0].nodeValue);
    document.write("<br>");
}
</script>
</body>
</html>

```

Execution

Save this file as *replacenode_example.htm* on the server path *thisfileandnode.xml* should be on the same path in your server. In the output we get the following:

```

Content of FirstName element before replace operation
Tanmay
Taniya
Tanisha

Content of FirstName element after replace operation
Taniya
Tanisha

```

replaceData

The method `replaceData` replaces the characters starting at the specified 16-bit unit offset with the specified string.

Syntax

The `replaceData` has following syntax:

```
void replaceData(int offset, int count, java.lang.String arg) throws DOMException
```

Where

- *offset*: is the offset from which to start replacing.
- *count*: is the number of 16-bit units to replace. If the sum of offset and count exceeds length, then all 16-bit units to the end of the data are replaced.
- *arg*: the *DOMString* with which the range must be replaced.

Example

The following example *replacedata_example.htm* parses an XML document ("[node.xml](#)") into an XML DOM object and replaces .

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function loadXMLDoc(filename)
      {
        if (window.XMLHttpRequest)
        {
          xhttp = new XMLHttpRequest();
        }
        else // code for IE5 and IE6
        {
          xhttp = new ActiveXObject("Microsoft.XMLHTTP");
        }
        xhttp.open("GET", filename, false);
        xhttp.send();
        return xhttp.responseXML;
      }
    </script>
  </head>
  <body>
    <script>
      xmlDoc = loadXMLDoc("/dom/node.xml");

      x = xmlDoc.getElementsByTagName("ContactNo")[0].childNodes[0];
      document.write("<b>ContactNo before replace operation:</b> "+x.nodeValue);
      x.replaceData(1, 5, "9999999");
      document.write("<br>");
      document.write("<b>ContactNo after replace operation:</b> "+x.nodeValue);

    </script>
  </body>
</html>
```

In the above example:

- *x.replaceData(2, 3, " 999 ")* :: Here x holds the text of the specified element <ContactNo> whose text is replaced by the new text "9999999", starting from the position 1 till the length of 5.

Execution

Save this file as *replacedata_example.htm* on the server path
thisfileandnode.xmlshouldbeonthesamepathinyourserver. In the output we get the following:

ContactNo before replace operation: 1234567890

ContactNo after replace operation: 199999997890

Processing math: 100%