

Laborator 08 - Scrierea unui document XML (PHP->XML),

Cuprins

[XMLWriter](#)

[Exemplu de utilizare a XMLWriter pentru a genera un document XML \(abordare procedurală\)](#)

[Exemplu de utilizare a XMLWriter pentru a genera un document XML \(abordare orientată pe obiecte\)](#)

[Utilizarea DOM pentru scrierea de documente](#)

[Exemplu de utilizare a DOM pentru a genera un document XML nou](#)

[Exemplu de utilizare a DOM pentru a adăuga date la un document XML existent](#)

În laboratoarele precedente am studiat modul în care pot fi citite datele dintr-un document XML (fie el stocat într-un string sau într-un fișier). Laboratorul de față conține diverse abordări ale scrierii datelor într-un document XML, la rândul lui stocat într-un fișier pe disc.

XMLWriter

XMLWriter este o bibliotecă PHP care conține elementele necesare pentru a genera un document XML care poate fi ulterior salvat într-un fișier pe disc sau utilizat în alt mod. XMLWriter poate fi folosită atât pentru o abordare procedurală cât și pentru una orientată pe obiecte.

Obs. XMLWriter este mai puțin potrivit pentru adăugarea de date la un fișier XML deja existent. Pentru o astfel de utilizare este recomandabil DOM.

Următoarele trei exemple vor genera trei documente XML având conținut identic folosind XMLWriter cu o abordare procedurală, XMLWriter cu o abordare orientată pe obiecte și DOM.

Exemplu de utilizare a XMLWriter pentru a genera un document XML (abordare procedurală)

```
<?php
echo "Încep scrierea unui document XML. <br />...<br />";
//Creaza resursa de tip XMLWriter
$xml=xmlwriter_open_memory();
//Setari legate de formatare. Fara ele documentul XML este scris pe o singura linie si e mai greu de citit
"cu ochiul liber"
xmlwriter_set_indent($xml,true);
xmlwriter_set_indent_string($xml," ");
//Creaza inceputul de document XML
xmlwriter_start_document($xml,'1.0', 'UTF-8');
//Creaza "radacina" unei serii de noduri
xmlwriter_start_element($xml,"proiecte");
//Creaza primul nod. Nu uit sa deschid si sa inchid fiecare element al nodului
xmlwriter_start_element($xml,"proiect");
    xmlwriter_write_attribute($xml,"id", 1);
    xmlwriter_start_element($xml,"data_predarii");
        xmlwriter_text($xml,date("d/m/Y h:i:s"));
    xmlwriter_end_element($xml);
    xmlwriter_start_element($xml,"autor");
```

```

        xmlwriter_write_attribute($xml,"functie", "student");
        xmlwriter_text($xml,"Ionescu Ionica");
    xmlwriter_end_element($xml);
xmlwriter_end_element($xml);
//Creaza al doilea nod nod
xmlwriter_start_element($xml,"proiect");
    xmlwriter_write_attribute($xml,"id", 2);
    xmlwriter_start_element($xml,"data_predarii");
        xmlwriter_text($xml,date("d/m/Y h:i:s"));
    xmlwriter_end_element($xml);
    xmlwriter_start_element($xml,"autor");
        xmlwriter_write_attribute($xml,"functie", "student");
        xmlwriter_text($xml,"Marinescu Marinica");
    xmlwriter_end_element($xml);
xmlwriter_end_element($xml);
//Inchid seria de noduri
xmlwriter_end_element($xml);
//Salvare completa. se foloseste la sfarsit
file_put_contents('proiecte2.xml', xmlwriter_output_memory ($xml));
echo "Am terminat scrierea documentului XML. <br />";

//sau ...

```

//Salvare partiala. Se foloseste pe parcurs de oricate ori este nevoie pentru a evita supraincercarea memoriei

```

//file_put_contents('proiecte.xml', xmlwriter_flush($xml,true), FILE_APPEND);
?>

```

Exemplu de utilizare a XMLWriter pentru a genera un document XML (abordare orientată pe obiecte)

```

<?php
echo "Incep scrierea unui document XML. <br />...<br />";
//Creaza un obiect nou de tip XMLWriter
$xml=new XMLWriter();
//Aloca memorie obiectului
$xml->openMemory();
//Setari legate de formatare. Fara ele documentul XML este scris pe o singura linie si e mai greu de citit
"cu ochiul liber"
$xml->setIndent(true);
$xml->setIndentString(" ");
//Creaza inceputul de document XML
$xml->startDocument('1.0', 'UTF-8');
    //Creaza "radacina" unei serii de noduri
    $xml->startElement("proiecte");
        //Creaza primul nod. Nu uit sa deschid si sa inchid fiecare element al nodului
        $xml->startElement("proiect");
            $xml->writeAttribute("id", 1);
            $xml->startElement("data_predarii");

```

```

        $xml->text(date("d/m/Y h:i:s"));
$xml->endElement();
$xml->startElement("autor");
        $xml->writeAttribute("functie", "student");
        $xml->text("Ionescu Ionica");
$xml->endElement();
$xml->endElement();
//Creaza al doilea nod nod
$xml->startElement("proiect");
        $xml->writeAttribute("id", 2);
        $xml->startElement("data_predarii");
                $xml->text(date("d/m/Y h:i:s"));
        $xml->endElement();
        $xml->startElement("autor");
                $xml->writeAttribute("functie", "student");
                $xml->text("Marinescu Marinica");
        $xml->endElement();
$xml->endElement();
//Inchid seria de noduri
$xml->endElement();
//Salvare completa. se foloseste la sfarsit
file_put_contents('proiecte1.xml', $xml->outputMemory());
echo "Am terminat scrierea documentului XML. <br />";

//sau ...

//Salvare partiala. Se foloseste pe parcurs de oricate ori este nevoie pentru a evita supraincercarea
memoriei
//file_put_contents('proiecte.xml', $xml->flush(true), FILE_APPEND);
?>

```

Utilizarea DOM pentru scrierea de documente

Biblioteca DOM, care a mai fost prezentată în laboratorul precedent (la momentul respectiv pentru citirea unui document) poate fi utilizată atât pentru generarea unui document XML nou cât și pentru adăugarea de date la un document XML existent. Următoarele două scripturi PHP vor exemplifica aceste două situații.

Exemplu de utilizare a DOM pentru a genera un document XML nou

```

<?php
echo "Incep scrierea unui document XML. <br />...<br />";
//Creaza un obiect nou de tip DOMDocument
$dom=new DOMDocument('1.0', 'UTF-8');
//Setari legate de formatare. Fara ele documentul XML este scris pe o singura linie si e mai greu de citit
"cu ochiul liber"
$dom->preserveWhiteSpace = false;
$dom->formatOutput=true;
//Creaza "radacina" unei serii de noduri
$proiecte=$dom->createElement("proiecte");
        //Creaza primul nod. Nu uit sa deschid si sa inchid fiecare element al nodului

```

```

$proiect=$dom->createElement("proiect");
    $proiect->setAttribute("id", 1);
    $published=$dom->createElement("data_predarii");
        $publishedDate=$dom->createTextNode(date("d/m/Y h:i:s", strtotime("now")));
        $published->appendChild($publishedDate);
    $proiect->appendChild($published);
    $autor=$dom->createElement("autor");
        $autor->setAttribute("functie", "student");
        $autorName=$dom->createTextNode("Ionescu Ionica");
    $autor->appendChild($autorName);
    $proiect->appendChild($autor);
$proiecte->appendChild($proiect);
//Creaza al doilea nod nod
$proiect=$dom->createElement("proiect");
    $proiect->setAttribute("id", 2);
    $published=$dom->createElement("data_predarii");
        $publishedDate=$dom->createTextNode(date("d/m/Y h:i:s", strtotime("now")));
        $published->appendChild($publishedDate);
    $proiect->appendChild($published);
    $autor=$dom->createElement("autor");
        $autor->setAttribute("functie", "student");
        $autorName=$dom->createTextNode("Marinescu Marinica");
    $autor->appendChild($autorName);
    $proiect->appendChild($autor);
$proiecte->appendChild($proiect);
$dom->appendChild($proiecte);
//Salvare sub un nume oarecare
$dom->save("proiecte3.xml");
echo "Am terminat scrierea documentului XML. <br />";
?>

```

Exemplu de utilizare a DOM pentru a adăuga date la un document XML existent

```

<?php
    echo "Incep adaugarea la un document XML. <br />...<br />";
    //Creaza un obiect nou de tip DOMDocument
    $dom=new DOMDocument('1.0', 'UTF-8');
    //Setari legate de formatare. Fara ele documentul XML este scris pe o singura linie si e mai greu de citit
    "cu ochiul liber"
    $dom->preserveWhiteSpace = false;
    $dom->formatOutput=true;
    //Incarc fisierul existent in obiectul creat mai sus
    $dom->load("proiecte3.xml");
    //Creaza un nod nou care se adauga la structura deja existenta. Nu uit sa deschid si sa inchid fiecare
    element al nodului
    $proiect=$dom->createElement("proiect");
    $proiect->setAttribute("id", 3);
    $published=$dom->createElement("data_predarii");
        $publishedDate=$dom->createTextNode(date("d/m/Y h:i:s", strtotime("tomorrow")));

```

```
        $published->appendChild($publishedDate);
$proiect->appendChild($published);
$autor=$dom->createElement("autor");
    $autor->setAttribute("functie", "student");
    $autorName=$dom->createTextNode("Gheorghiu Gheorghe");
$autor->appendChild($autorName);
$proiect->appendChild($autor);
$dom->documentElement->appendChild($proiect);
//Salvare sub un nume oarecare
$dom->save("proiecte4.xml");
echo "Am terminat adaugarea la documentul XML. <br />";
?>
```