

Laboration 7

I denna lab ska vi exportera produkterna i tabellerna products i olika format. Först som en vanlig html-sida, sedan som en tabseparerad fil och slutligen i xml-format.

Uppgift 1

Uppgiften är att söka alla produkter i tabellen products, och visa en rubrik med produktnamn och pris, sedan en bild, sedan produktbeskrivning. Ni har tillgång till en zip-fil med en bild för varje produkt, och filnamnet är *prod_id.jpg*. För tex Bird bean bag toy med produktid *BNBG02* är filnamnet på bilden *BNBG02.jpg*.

Webbsidan kan till exempel se ut så här:

All products

Fish bean bag toy \$3.49



Fish bean bag toy, complete with bean bag worms with which to feed it

Bird bean bag toy \$3.49



Bird bean bag toy, eggs are not included

Rabbit bean bag toy \$3.49



Uppgift 2

Uppgiftern är att skriva en php-sida som skapar en tab-separerad fil med kolumnerna prod_id, prod_price, prod_name och prod_desc för alla produkter i tabellen products. Sidan ska med funktionen file_put_contents() skapa en textfil samt visa en länk till filen. Sidan kan se ut så här:

A file has been created

[export.txt](#)

När man klickar export.txt ska man få fram den tab-separerade filen:

```
BNBG01 3.49 Fish bean bag toy Fish bean bag toy, complete with bean bag v
BNBG02 3.49 Bird bean bag toy Bird bean bag toy, eggs are not included
BNBG03 3.49 Rabbit bean bag toy Rabbit bean bag toy, comes with bean bag ca
BR01 5.99 8 inch teddy bear 8 inch teddy bear, comes with cap and jacke
BR02 8.99 12 inch teddy bear 12 inch teddy bear, comes with cap and jacke
BR03 11.99 18 inch teddy bear 18 inch teddy bear, comes with cap and jacke
RGAN01 4.99 Raggedy Ann 18 inch Raggedy Ann doll
RYL01 9.49 King doll 12 inch king doll with royal garments and crown
RYL02 9.49 Queen doll 12 inch queen doll with royal garments and crown
```

Uppgift 3

Nu ska istället en xml-fil skapas med samma data som i tab-filen i förra uppgiften. Målet är att filen ska ha följande struktur:

```
▼<toycatalog>
  ▶<toy>...</toy>
  ▼<toy>
    <prod_id>BNBG02</prod_id>
    <prod_price>3.49</prod_price>
    <prod_name>Bird bean bag toy</prod_name>
    <prod_desc>Bird bean bag toy, eggs are not included</prod_desc>
  </toy>
  ▶<toy>...</toy>
  ▶<toy>...</toy>
  ▶<toy>...</toy>
  ▶<toy>...</toy>
  ▶<toy>...</toy>
  ▶<toy>...</toy>
  ▶<toy>...</toy>
</toycatalog>
```

Rotelementet heter alltså <toycatalog>. Varje rad i tabellen products motsvaras av elementet <toy>. Varje <toy>...</toy> innehåller i sin tur elementen <prod_id>, <prod_price>, <prod_name> och <prod_desc>.

Gör en php-sida som skapar xml-filen *export1.xml* och visar en länk till den:

A file has been created

[export1.xml](#)

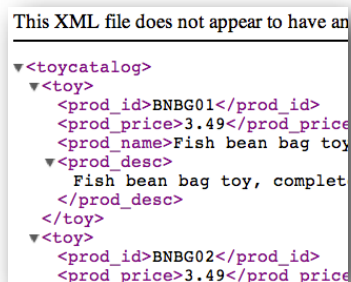
Börja med en variabel som tex heter \$export och får rotelementet som startvärde:

```
$export="<toycatalog>";
```

bygg sedan på denna variabel i en while-loop som loopar igenom resultatet från sökningen i products:

```
$export.="<toy>";
$export.="<prod_id>";
$export.= här ska ett värde ut tabellen products komma
$export.="</prod_id>";
etc
```

När man klickar länken ska xml-sidan visas:



Uppgift 4 - frivillig

PHP har ett antal färdiga funktioner för att bygga upp xml-dokument.

Ett rotelement med namnet *toycatalog* kan skapas med följande rader kod:

```
$doc = new DomDocument('1.0');
$root = $doc->createElement('toycatalog');
$root = $doc->appendChild($root);
```

För varje rad i tabellen skapas ett *toy*-element med följande rader:

```
$rownode = $doc->createElement('toy');
$root->appendChild($rownode);
```

För varje kolumn skapas ett element enligt följande exempel för kolumnen *prod_id*:

```
//här skapas taggen <prod_id>:
$child = $doc->createElement("prod_id");
$rownode->appendChild($child);
//här fylls taggen <prod_id> med innehåll:
$value = $doc->createTextNode($row["prod_id"]);
$child->appendChild($value);
```

När alla rader och kolumner är tillagda finns hela xml-strukturen i variabeln `$doc`. Denna variabel kan dock inte skrivas ut direkt utan måste konverteras till en sträng som sedan kan exporteras i en fil. Det görs med koden

```
$export = $doc->saveXML();  
//$export innehåller nu all xml som en text
```

Elementet får namnet *prod_id* och innehåller det som finns i `$row["prod_id"]`, dvs det värde som hämtats ur databasen för aktuell rad och kolumn.

Skapa en php-sida som i 3 men som använder funktionerna ovan för att skapa själva xml-dokumentet.