

Abstract

In the following article, we will discuss the implementation of various web data modules using php3 and Interbase.

It is assumed that there is a web server (e.g apache) with php integration and that those php modules are compiled with interbase support.

Specifically, we will discuss the following:

- a) Preparation of the connection to an interbase server using php
- b) Creation and execution of dynamic SQL statements
- c) Interbase Blob field handling using php
- d) Acquiring and processing values from forms

Introduction

PHP is a powerful server side script language, which supports almost every Known SQL Database Server. But the real strength of PHP, however is that it has some astonishing characteristics in the string manipulation field as also some really flexible graphics processing features.

In the examples that follow we will assume that there is an Interbase DB Server at the TCP/IP address 192.168.0.1 and a database named main.gdb. The main.gdb contains the following tables yp_foreis, ypotrofies_eidi, ypotr. Theoretically these tables contain information on scholarships. In our case Ypotr table is presumed to be the main scholarship table with the following data structure.

```
CREATE TABLE NEW_YPOTR (AA DOUBLE PRECISION NOT NULL,  
□  
    KATAGOGI VARCHAR(100),  
    SKOPOS SMALLINT,  
    APAITISEIS BLOB SUB_TYPE TEXT SEGMENT SIZE 80,  
    ANTIKEIMENO_SPOYDON BLOB SUB_TYPE TEXT SEGMENT SIZE 80,  
    TOPOS_SPOYDON VARCHAR(200),  
    DIARKEIA SMALLINT,  
    KRITIRIA BLOB SUB_TYPE TEXT SEGMENT SIZE 80,  
    TOTAL_VALUE DOUBLE PRECISION,  
    PAROXES BLOB SUB_TYPE TEXT SEGMENT SIZE 80,  
    HMEROMINIA DATE,  
    DIKAIOLOGITIKA BLOB SUB_TYPE TEXT SEGMENT SIZE 80,  
    PLHROFORIES BLOB SUB_TYPE TEXT SEGMENT SIZE 80,  
    FOREAS_ID DOUBLE PRECISION,  
PRIMARY KEY (AA));
```

The Yp_Foreis table below is a details table linked with the master at the fields AA->foreas_id and contains information about the conveyor of the scholarship.

```
CREATE TABLE YP_FOREIS (AA DOUBLE PRECISION NOT NULL,  
    TITLE VARCHAR(250),  
PRIMARY KEY (AA));
```

□
Ypotrofies_eidi table is a details table linked with the master at the fields AA->skopos and contains information about the aim of the scholarship.

```
CREATE TABLE YPOTROFIES_EIDI (AA SMALLINT NOT NULL,  
  
    TITLE VARCHAR(250),  
PRIMARY KEY (AA));
```

The goal of the web data application we will try to construct, is to provide a web interface which follows the QBE paradigm and which will enable a user to search the scholarship database for a scholarship which matches the criteria he/she enters in a web form.

The total application consists of 3 php modules so that as a whole the application passes from 3 distinct phases.

- 1) The first phase is the creation of a form with 2 combo-boxes filled dynamically with the entire information of the previous 2 details tables. In this phase the php module contains the necessary code for an interbase connection, and traces those tables from the beginning to the end exporting values to the appropriate php variables.
- 2) The second module is responsible for the gathering of the information the user enters, the dynamic transformation of the SQL queries and the translation of there results in a generic three column web table. The information that is presented in this table consists of an auto-numbered column, a column with the name of the conveyor of the scholarship and an href tag leading to more detailed description of the scholarship and a column with some generic information of the scholarship as it comes from the field antikeimeno_spydon of the master table.
- 3) The third php module is responsible to transform a selected href tag to an appropriate page with form layout containing detailed description of the selected scholarship.

The modules

Module1 action1.php3

```

1  <html>
2  □
3
4
5  <head>
6  <title>ΦΟΡΜΑ ΑΝΑΖΗΤΗΣΗΣ ΥΠΟΤΡΟΦΙΩΝ</title>
7  <meta name="GENERATOR" content="joe">
8  <base href="http://career.admin.uoi.gr/">
9  </head>
10
11 <body link="#000000" vlink="#000000" alink="#000000" style="text-
12 decoration:" none
13 background="back-white.jpg">
14 <div align="center"><center>
15
16 <table border="0" cellpadding="0" cellspacing="0" style="border:" 2px
17 none rgb(0,64,128)
18 width="530">
19   <tr>
20     <td width="100%"><div align="center"><center><table border="0"
21 cellpadding="0"
22 cellspacing="0" height="38" width="100%">
23   <tr>
24     <td height="38" width="100%"><p align="center"><font
25 color="#000080"><strong>ΦΟΡΜΑ&nbsp;
26 ΑΝΑΖΗΤΗΣΗΣ &nbsp; ΥΠΟΤΡΟΦΙΩΝ</strong></font></td>
27   </tr>
28 </table>
29 </center></div><hr>
30 <form method="GET" action="/ypo2.php3">
31 <p><strong>Κριτήρια αναζήτησης:</strong></p>
32 <table border="1" width="100%">
33   <tr>
34     <td width="50%">Φορέας Υποτροφίας:&nbsp;</td>
35     <td width="50%"><select name="foreas" size="1">
36       <option value="*">*</option>
37 <?php
38 $base="192.168.0.1:/main.gdb";
39 $i=ibase_connect($base,"SYSDBA","masterkey");
40 $q=ibase_prepare("select aa,title from yp_foreis order by title");
41 $result=ibase_execute($q,1);
42 while ($row=ibase_fetch_object($result))

```



```
104     </tr>
105 </table>
106 </center></div>
107 </body>
108 </html>
```

The php code which performs the preparation, the connection and the tracing of the table is on lines 35-44 and 56-66. Lets see this code step by step.

```
1 <?php
2 □
3 $base="192.168.0.1:/main.gdb";
4 □
5 $i=ibase_connect($base,"SYSDBA","masterkey");
6 □
7 $q=ibase_prepare("select aa,title from ypotrofies_eidi order by
8 title");
9 □
10 $result=ibase_execute($q,1);
11 □
12 while ($row=ibase_fetch_object($result))
13 □
14 echo "<option value=", $row->AA , "> ",substr($row->TITLE,0,40) ,
15 "</option>";
16 ibase_close();
17 ?>
```

In line No 2 a variable named “base” is created and a value of a standard TCP/IP database address is placed in it.

In line No 3 the `ibase_connect` function is called with three parameters. These are:

The previous base variable with the name of the database, a string with the name of the user, in this case “SYSDBA”, and a string with the name of the password in this case “masterkey”. The returned value fills the variable `i` in such case that we would like to perform a detailed error handling. In our case no error handling is performed and if an error occurs the standard web server error response will be returned to the user.

In line No 4 an established connection with a database already exists and the `ibase_prepare` function is being called with a static SQL statement as a string parameter. This statement is used in order to allow the transformation of the SQL statement into an internal format which interbase understands. Of course, as you can see lines 3 and 4 could be combined into the following one line statement `ibase_execute(ibase_prepare("select aa,title from ypotrofies_eidi order by title"),1)`; however the previous two line method is more readable and also permits error handling in case the “q” parameter has a null value. After line 6 has been executed, we have a result-set returned to the variable “result” with a cursor positioned at the top of the file. At this point it is necessary to start a loop using function `ibase_fetch_object`, which in turn performs the tracing of the result-set record by record till the end of the result-set. Meanwhile the values of the table columns named “aa”, “title” are joined to create an option html tag. The transformation of the double precision value of “aa” to a string is performed automatically by php.

In line 10 we have ended filling our combo-box and the connection to an interbase server is no longer needed. Therefore, we call function `ibase_close` to terminate the connection.

In line 28 of the whole html page “<form method="GET" action="/action2.php3">” the action that is being taken when the user submits the form is to execute then next php page.

Module1 action2.php3

```
1 <html>
2 □
3
4 □
5 <head>
6 □
7 <title>ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ</title>
8 <base href="http://career.admin.uoi.gr/">
9 </head>
10
11 <body link="#000000" vlink="#000000" alink="#000000" style="text-
12 decoration: none"
13 background="back-white.jpg">
14 <div align="center"><center>
15
16 <table border="0" width="99%">
17 <tr>
18 <td width="100%"><p align="right">&nbsp;</p>
19 <p align="center"><font face="Arial" size="2"
20 color="#000080"><strong>ΑΠΟΤΕΛΕΣΜΑΤΑ
21 ΑΝΑΖΗΤΗΣΗΣ ΥΠΟΤΡΟΦΙΩΝ<br>
22 </strong></font></p>
23 <div align="center"><center><table border="1" width="100%">
24 <tr>
25 <td width="4%" align="center"><p align="left"><font
26 face="Arial" size="2" color="#EC7600"><strong>&nbsp;<br>
27 A/A</strong></font></td>
28 <td width="42%" align="center"><font face="Arial" size="2"
29 color="#EC7600"><strong>Φορέας
30 Υποτροφίας*</strong></font></td>
31 <td width="\54%"><font face="Arial" size="2"
32 color="#EC7600"><strong>Αντικείμενο Σπουδών
33 </strong></font></td>
34 </tr>
35 <tr>
36 <?php
37
38 $sql1="select NEW_YPOTR.aa as aa,YP_FOREIS.title as title,
39 new_ypotr.antikeimeno_spoymdon from new_ypotr left join yp_foreis on
40 new_ypotr.foreas_id=yp_foreis.aa where %param1% and %param2% and
41 %param3% and %param4% order by foreas_id";
42
43 if ($foreas=="")
44 $sql1=str_replace("%param1%"," 0=0 ",$sql1);
45 else
46 $sql1=str_replace("%param1%"," foreas_id=$foreas ",$sql1);
47
48
49 if ($title1=="")
50 $sql1=str_replace("%param2%"," 0=0 ",$sql1);
51 else
52 $sql1=str_replace("%param2%"," katagogi like \"%$title1%\" ",$sql1);
53
54 if ($D1=="")
55 $sql1=str_replace("%param3%"," 0=0 ",$sql1);
56 else
57 $sql1=str_replace("%param3%"," skopos=$D1 ",$sql1);
58
59 if ($title2=="")
```

```

60 $sql1=str_replace("%param4%", " 0=0 ", $sql1);
61 else
62 $sql1=str_replace("%param4%", " (ANTIKEIMENO_SPOYDON like
63 \"%$title2%\" or ANTIKEIMENO_SPOYDON like \"%ΔΕΝ ΥΠΑΡΧΕΙ
64 ΠΕΡΙΟΡΙΣΜΟΣ%\" )", $sql1);
65
66 $j=1;
67 $base="192.168.0.1:/main.gdb";
68 $i=ibase_connect($base, "SYSDBA", "masterkey");
69 $result=ibase_query($sql1);
70 while ($row=ibase_fetch_row($result, IBASE_TEXT+
71 IBASE_TEXT+IBASE_TEXT))
72 {
73 echo " <tr> <td width=\"4%\" align=left>&nbsp;",$j,"</td><td
74 width=\"42%\"><align=left><a
75 href=http://career.admin.uoi.gr/ypo4.php3?",$row[0],">",$row[1], "</a>
76 </td><td width=\"54%\">",$row[2], " </td></tr>";
77 $j=$j+1;
78 }
79 ibase_free_result($result);
80 ibase_close();
81 ?>
82
83 <tr>
84 </table>
85 </center></div><p>* Για περισσότερες πληροφορίες κάντε κλίκ στον
86 τίτλο της Υποτροφίας. </td>
87 </tr>
88 </table>
89 </center></div>
90 </body>
91 </html>

```

In this module we will learn how to create and execute a dynamic sql statement.

The php code that does this is at lines 33-85. The technique is based on the use of the php function named “str_replace”, an almost static SQL statement and a trick. We say “almost static” because as you can see, (lines 33-38 above and below for your convenience) contains 4 substrings named from param1 to param4 and placed between the percent symbol (%). This is not a php symbol with special meaning. It is a symbol of our choice to make it easier for us to search for it and replace it with another string according to the state of some conditions. The various conditions are at lines 40 to 69 above. In more detail, our goal is to find a way to change the following statement :

```

“$sql1="select NEW_YPOTR.aa as aa,YP_FOREIS.title as title,
new_ypotr.antikeimeno_spoymdon from new_ypotr left join yp_foreis on
new_ypotr.foreas_id=yp_foreis.aa where %param1% and %param2% and
%param3% and %param4% order by foreas_id";”

```

from something like this

```

“$sql1="select NEW_YPOTR.aa as aa,YP_FOREIS.title as title,
new_ypotr.antikeimeno_spoymdon from new_ypotr left join yp_foreis on
new_ypotr.foreas_id=yp_foreis.aa order by foreas_id";”

```

which means select everything,

to something like this:

```

“$sql1="select NEW_YPOTR.aa as aa,YP_FOREIS.title as title,
new_ypotr.antikeimeno_spoymdon from new_ypotr left join yp_foreis on
new_ypotr.foreas_id=yp_foreis.aa where foreas_id=99 and katagogi like

```

```
"%something%" and skopos=3 and ANTIKEIMENO_SPOYDON like "%something else%" order by foreas_id";"
```

meaning select those records which match the following criteria:

```
foreas_id=99 and katagogi like "something" and skopos=3 and ANTIKEIMENO_SPOYDON like "%something else%"
```

As we already mentioned this is done by combining the `str_replace` php function and the trick to replace those variables which are empty with an always true condition `"0=0"`

Analyzing lines 40-43 above we point out the following:

The `"foreas"` php variable is the variable which contains the user selection in the first combo-box in the first module. This variable depending on the user choice, can contain various values in the range from `"*"`, meaning everything to every other distinct value as filled from the `ypotrofies_eidi` table after the execution of the `action1` php module. (The first php module). Therefore in the case that the user selects `"*"` the substring `%param1%` must be replaced with an always true condition, in other words the `"0=0"` value. In any other case the substring `"%param1%"` must yield in something like `"foreas_id=19"` where 19 is the `aa` field value from the `ypotrofies_eidi` table at the position the user made his/her choice in the combo-box.

For your convenience we repeat lines 40-43 below.

```
if ($foreas=="*")
□
$sql1=str_replace("%param1%"," 0=0 ",$sql1);
□
else
□
$sql1=str_replace("%param1%"," foreas_id=$foreas ",$sql1);
□
```

The remaining php code is quite trivial, and some of the php functions related to interbase are already described in module1. The new function here is the `ibase_query` function that is a way to prepare and executes an SQL statement in only one line of code. Another function is the `ibase_free_result` function, which is called to dispose any reserved memory after the execution of an SQL statement and the handling of the returned results.

The row creation is on line 77-80 and we believe it is quite understandable and no explanation needed.

Module action3.php3

```
1 <html>
2 □
3
4 □
5 <head>
6 □
7 <title>ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ</title>
8 <base href="http://career.admin.uoi.gr/">
9 </head>
10
11 <body link="#000000" vlink="#000000" alink="#000000" style="text-
12 decoration: none"
13 background="back-white.jpg">
14 <?php
15 $base="192.168.0.1:/main.gdb";
16 $i=ibase_connect($base,"SYSDBA","masterkey");
17 ibase_close();
18 $base="192.168.0.1:/main.gdb";
19 $i=ibase_connect($base,"SYSDBA","masterkey");
20 $q=ibase_prepare("select * from new_ypotr where aa=$QUERY_STRING");
21 $result=ibase_execute($q,1);
22 $row=ibase_fetch_object($result);
23 $sql1="select * from yp_foreis where aa=$row->FOREAS_ID";
24 $q1=ibase_prepare($sql1);
25 $result1=ibase_execute($q1,1);
26 $row1=ibase_fetch_object($result1);
27
28 $q2=ibase_prepare("select * from ypotrofies_eidi where aa=$row-
29 >SKOPOS");
30 $result2=ibase_execute($q2,1);
31 $row2=ibase_fetch_object($result2);
32
33 $blob_id = ibase_blob_open($row->ΑΠΑΙΤΗΣΕΙΣ);
34 if ($blob_id!=0)
35 { $apaitiseis_string = ibase_blob_get($blob_id,1000);
36 }
37 $blob_id = ibase_blob_open($row->ΑΝΤΙΚΕΙΜΕΝΟ_ΣΠΟΥΔΩΝ);
38 if ($blob_id!=0)
39 { $antikeimeno_string = ibase_blob_get($blob_id,1000);}
40
41 $blob_id = ibase_blob_open($row->ΚΡΙΤΗΡΙΑ);
42 if ($blob_id!=0)
43 { $kritiria_string = ibase_blob_get($blob_id,1000);}
44
45 $blob_id = ibase_blob_open($row->ΠΑΡΟΧΕΣ);
46 if ($blob_id!=0)
47 { $paroxes_string = ibase_blob_get($blob_id,1000);}
48
49 $blob_id = ibase_blob_open($row->ΔΙΚΑΙΟΛΟΓΙΤΙΚΑ);
50 if ($blob_id!=0)
51 { $dikaiologitika_string = ibase_blob_get($blob_id,1000);}
52
53 $blob_id = ibase_blob_open($row->ΠΛΗΡΟΦΟΡΙΕΣ);
54 if ($blob_id!=0)
55 { $plirofories_string = ibase_blob_get($blob_id,1000);}
56
57
58 ibase_close();
```

```

59  ?>
60
61  <div align="center"><center>
62
63  <table border="0" width="93%">
64    <tr>
65      <td width="100%"><p align="center"><font face="Arial" size="2"
66  color="#000080"><strong>ΠΑΡΟΥΣΙΑΣΗ
67  ΥΠΟΤΡΟΦΙΑΣ</strong></font></p>
68    <hr width="80%">
69    <p><strong><font face="Arial" size="2" color="#000080"><br>
70  ΦΟΡΕΑΣ ΥΠΟΤΡΟΦΙΑΣ</font></strong></p>
71    <div align="center"><center><table border="1" width="100%">
72      <tr>
73        <td width="100%" bgcolor="#FFFFFF">&nbsp;
74  <?php
75  echo $row1->TITLE
76  ?>
77  </td>
78    </tr>
79    </table>
80    </center></div><p><font face="Arial" size="2"
81  color="#000080"><strong>ΚΑΤΑΓΩΓΗ
82  ΑΠΟΔΕΚΤΩΝ</strong></font></p>
83    <div align="center"><center><table border="1" width="100%">
84      <tr>
85        <td width="100%" bgcolor="#FFFFFF">&nbsp;
86  <?php
87  echo $row->KATAGOGE
88  ?>
89  </td>
90    </tr>
91    </table>
92    </center></div><p><strong><font face="Arial" size="2"
93  color="#000080">ΣΚΟΠΟΣ </font></strong></p>
94    <div align="center"><center><table border="1" width="100%">
95      <tr>
96        <td width="100%" bgcolor="#FFFFFF">&nbsp;
97  <?php
98  echo $row2->TITLE
99  ?>
100   </td>
101   </tr>
102   </table>
103   </center></div><p><strong><font face="Arial" size="2"
104  color="#000080">ΑΠΑΙΤΟΥΜΕΝΑ
105  ΣΤΟΙΧΕΙΑ ΑΠΟΔΕΚΤΩΝ</font></strong></p>
106   <div align="center"><center><table border="1" width="100%">
107     <tr>
108       <td width="100%" bgcolor="#FFFFFF">&nbsp;
109   <?php
110   echo $apaitiseis_string
111   ?>
112   </td>
113     </tr>
114     </table>
115     </center></div><p><strong><font face="Arial" size="2"
116  color="#000080">ΑΝΤΙΚΕΙΜΕΝΟ
117  ΣΠΟΥΔΩΝ ΠΟΥ ΑΠΕΥΘΥΝΕΤΑΙ Η ΥΠΟΤΡΟΦΙΑ</font></strong></p>
118   <div align="center"><center><table border="1" width="100%">
119     <tr>

```

```

120         <td width="100%" bgcolor="#FFFFFF">&nbsp;
121     <?php
122     echo $antikeimeno_string
123     ?>
124 </td>
125     </tr>
126 </table>
127 </center></div><p><font face="Arial" size="2"
128 color="#000080"><strong>ΤΟΠΟΣ
129 ΣΠΟΥΔΩΝ</strong></font></p>
130 <div align="center"><center><table border="1" width="100%">
131 <tr>
132 <td width="100%" bgcolor="#FFFFFF">&nbsp;
133 <?php
134 echo $row->ΤΟΠΟΣ_SΠΟΥΔΩΝ
135 ?>
136 </td>
137 </tr>
138 </table>
139 </center></div><p><font face="Arial" size="2"
140 color="#000080"><strong>ΔΙΑΡΚΕΙΑ
141 ΥΠΟΤΡΟΦΙΑΣ ΣΕ ΜΗΝΕΣ</strong></font></p>
142 <div align="center"><center><table border="1" width="100%">
143 <tr>
144 <td width="100%" bgcolor="#FFFFFF">&nbsp;
145 <?php
146 echo $row->ΔΙΑΡΚΕΙΑ
147 ?>
148 </td>
149 </tr>
150 </table>
151 </center></div><p><font face="Arial" size="2"
152 color="#000080"><strong>ΚΡΙΤΗΡΙΑ
153 ΕΠΙΛΟΓΗΣ</strong></font></p>
154 <div align="center"><center><table border="1" width="100%">
155 <tr>
156 <td width="100%" bgcolor="#FFFFFF">&nbsp;
157 <?php
158 echo $kritiria_string
159 ?>
160 </td>
161 </tr>
162 </table>
163 </center></div><p><font face="Arial" size="2"
164 color="#000080"><strong>ΥΨΟΣ
165 ΥΠΟΤΡΟΦΙΑΣ</strong></font></p>
166 <div align="center"><center><table border="1" width="100%">
167 <tr>
168 <td width="100%" bgcolor="#FFFFFF">&nbsp;
169 <?php
170 echo $row->TOTAL_VALUE
171 ?>
172 </td>
173 </tr>
174 </table>
175 </center></div><p><font face="Arial" size="2"
176 color="#000080"><strong>ΕΠΙΠΛΕΟΝ
177 ΠΑΡΟΧΕΣ</strong></font></p>
178 <div align="center"><center><table border="1" width="100%">
179 <tr>
180 <td width="100%" bgcolor="#FFFFFF">&nbsp;

```

```

181 <?php
182 echo $paroxes_string
183 ?>
184 </td>
185 </tr>
186 </table>
187 </center></div><p><font face="Arial" size="2"
188 color="#000080"><strong>ΚΑΤΑΛΗΚΤΙΚΗ
189 ΗΜΕΡΟΜΗΝΙΑ ΥΠΟΔΟΛΗΣ ΔΙΚΑΙΟΛΟΓΗΤΙΚΩΝ</strong></font></p>
190 <div align="center"><center><table border="1" width="100%">
191 <tr>
192 <td width="100%" bgcolor="#FFFFFF">&nbsp;
193 <?php
194 echo $row->HMEROMINIA
195 ?>
196 </td>
197 </tr>
198 </table>
199 </center></div><p><font face="Arial" size="2"
200 color="#000080"><strong>ΑΛΛΑ
201 ΔΙΚΑΙΟΛΟΓΗΤΙΚΑ</strong></font></p>
202 <div align="center"><center><table border="1" width="100%">
203 <tr>
204 <td width="100%" bgcolor="#FFFFFF">&nbsp;
205 <?php
206 echo $dikaiologitika_string
207 ?>
208 </td>
209 </tr>
210 </table>
211 </center></div><p><font face="Arial" size="2"
212 color="#000080"><strong>ΑΛΛΕΣ
213 ΠΛΗΡΟΦΟΡΙΕΣ</strong></font></p>
214 <div align="center"><center><table border="1" width="100%">
215 <tr>
216 <td width="100%" bgcolor="#FFFFFF">&nbsp;
217 <?php
218 echo $plirofories_string
219 ?>
220
221 </td>
222 </tr>
223 </table>
224 </center></div><hr width="80%">
225 </td>
226 </tr>
227 </table>
228 </center></div>
229 </body>
230 </html>

```

In this module a creation of result page in form layout is being done. The main body of the php code is at lines from 11 to 56, but also in various places distinct php code shows up as a form layout design demands. The new function here is the `ibase_blob_open` and `ibase_blob_get`. These functions are really useful when we must handle blob interbase fields. The `ibase_blob_open` function makes the necessary memory allocation for the proper blob handling and the `ibase_blob_get` make blob to string conversions with predefined string size.

Conclusion

The above task if we tried to implement it in a form of embedded sql in a third language (e.g C), could probably expand our code in at least a range of 3 excluding the necessary html code. ¹

¹ This real working example can be found at <http://career.admin.uoi.gr>.