Accessing kdb+ from PHP to create dynamic webpages

John Ludlow

July 14, 2015

1 Introduction

Currently, a popular set-up for many dynamic websites involves a web server such Apache, the MySQL database and a scripting language such as PHP. PHP code is included in a HTML source file, and allows server-side processing, along with requests to MySQL databases, thus enabling dynamic websites. Here it is examined how PHP can be used to connect to a kdb+ server from a webpage. First it is shown how the kdb+ ODBC server can enable connections to be made from PHP that enables fine grained control over the returned data. This will demonstrated via a simple proof of principle example. As an extension, PHP is then used to submit a query to a kdb+ server using a web form with the result returned via an iframe using both a simple GET request and ODBC. The work presented in this report was carried out on a laptop running Windows Vista.
Note that a more efficient approach than ODBC to querying kdb+ from php is via a direct ipc connection, see the QPHP package:
https://github.com/geocar/qphp
and
https://github.com/tjcelaya/qphp

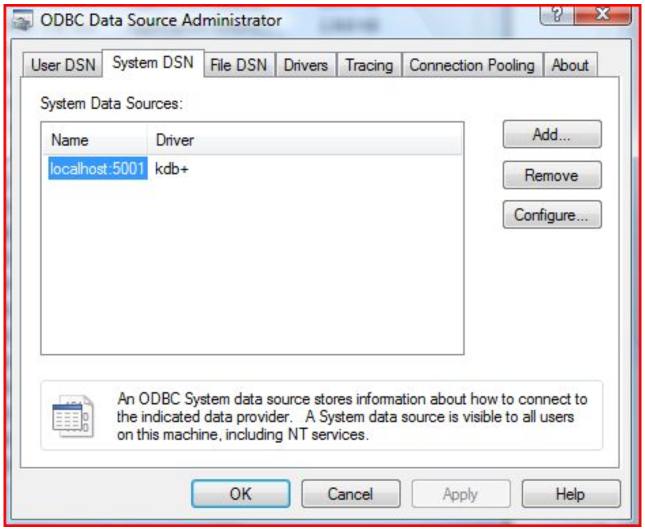
2 Connecting kdb+ and PHP using ODBC

Open Database Connectivity (ODBC) can be used as an interface for exchanging data between databases. To access data from non-kdb+ databases

from a kdb+ client, see:
https://code.kx.com/trac/wiki/Cookbook/ODBC/qclient
and
https://code.kx.com/trac/browser/contrib/jludlow/docs/odbc.pdf?format=raw
Here we will examine the alternative setup of kdb+ as the server, with a non-kdb+ client. Currently this is possible only for Windows. See:
https://code.kx.com/trac/wiki/Cookbook/ODBC/qserver
Details are presented for connecting from Excel and Visual Basic.

1. Download and run the appropriate kdb+ ODBC driver (32 or 64 bit).
   In this case for 32 bits:
   https://code.kx.com/trac/browser/kx/kdb+/w32/odbc.exe
   For 64 bits, download:
   https://code.kx.com/trac/browser/kx/kdb+/w64/odbc.zip
   and run d0.exe to install the ODBC driver.

2. Now a system DSN needs to be defined.
   Run C:\ Windows\System32\odbcas32.exe or C:\Windows\SysWOW64\odbcad32.exe depending on your operating system type.
   You will need to specify a host and port, in this case localhost and port 5001. Remember to set the correct system permissions for your q folder.



System Data Sources:

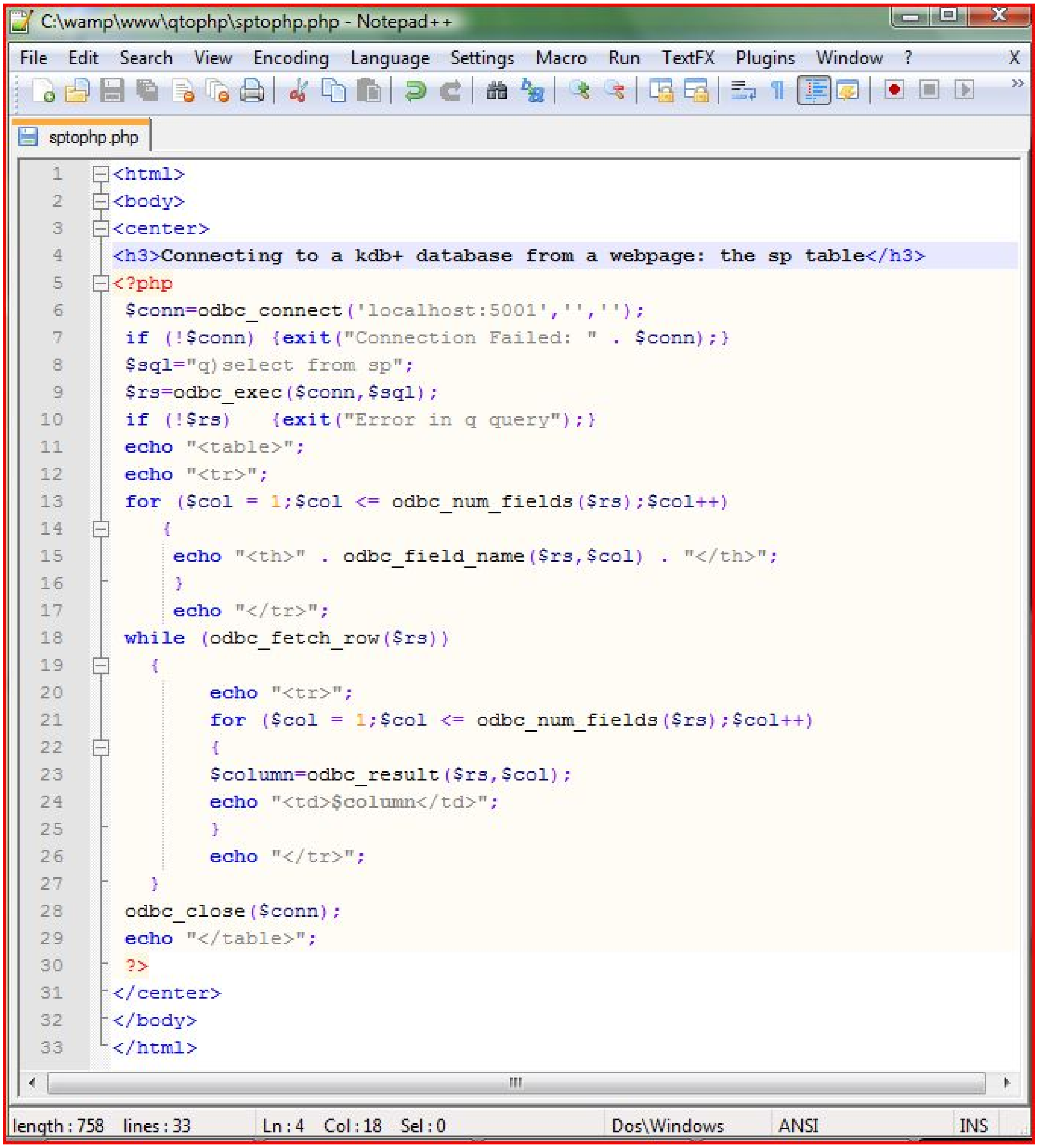
Name	Driver
localhost:5001	kdb+

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

3. Now we will install a web server and PHP. For Windows, an excellent free package is WampServer
   http://www.wampserver.com/en/download.php

4. For the example data we will make use of the Suppliers and Parts database that can be loaded into kdb+ via the sp.q script.
   https://code.kx.com/trac/wiki/Startingkdbplus/tables#a4.3SuppliersandParts

5. Now, we will write a simple PHP script to pull information from a kdb+ database via ODBC. For this we make use of the php ODBC connector:
   http://www.w3schools.com/PHP/php_db_odbc.asp


C:\wamp\www\qtophp\sptophp.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ? X
sptophp.php
length : 758 lines : 33 Ln : 4 Col : 18 Sel : 0 Dos\Windows ANSI INS

```
<html>
<body>
<center>
<h3>Connecting to a kdb+ database from a webpage: the sp table</h3>
<?php
 $conn=odbc_connect('localhost:5001','','');
 if (!$conn) {exit("Connection Failed: " . $conn);}
 $sql="q)select from sp";
 $rs=odbc_exec($conn,$sql);
 if (!$rs)   {exit("Error in q query");}
 echo "<table>";
 echo "<tr>";
 for ($col = 1;$col <= odbc_num_fields($rs);$col++)
    {
     echo "<th>" . odbc_field_name($rs,$col) . "</th>";
     }
     echo "</tr>";
 while (odbc_fetch_row($rs))
   {
        echo "<tr>";
        for ($col = 1;$col <= odbc_num_fields($rs);$col++)
        {
        $column=odbc_result($rs,$col);
        echo "<td>$column</td>";
        }
        echo "</tr>";
   }
 odbc_close($conn);
 echo "</table>";
 ?>
</center>
</body>
</html>
```

6. The php script sptophp.php will be saved in a subdirectory under C:\wamp\www
   Note that the ODBC connection is established via the line (no username or password is needed here):

```
$conn=odbc_connect('localhost:5001','','')
```

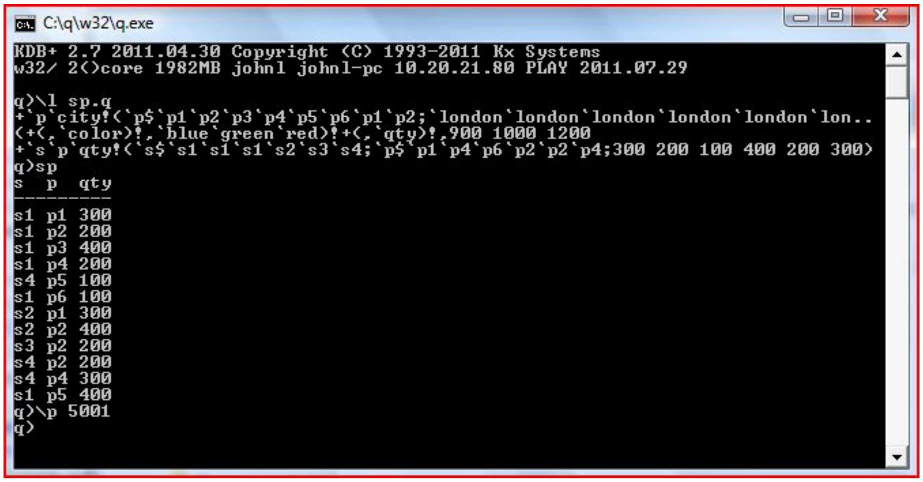
and the query as:

```
$sql="q)select from sp";
$rs=odbc_exec($conn,$sql);
```

Note that the query can also be written in standard SQL as:

```
$sql="s)select * from sp";
```

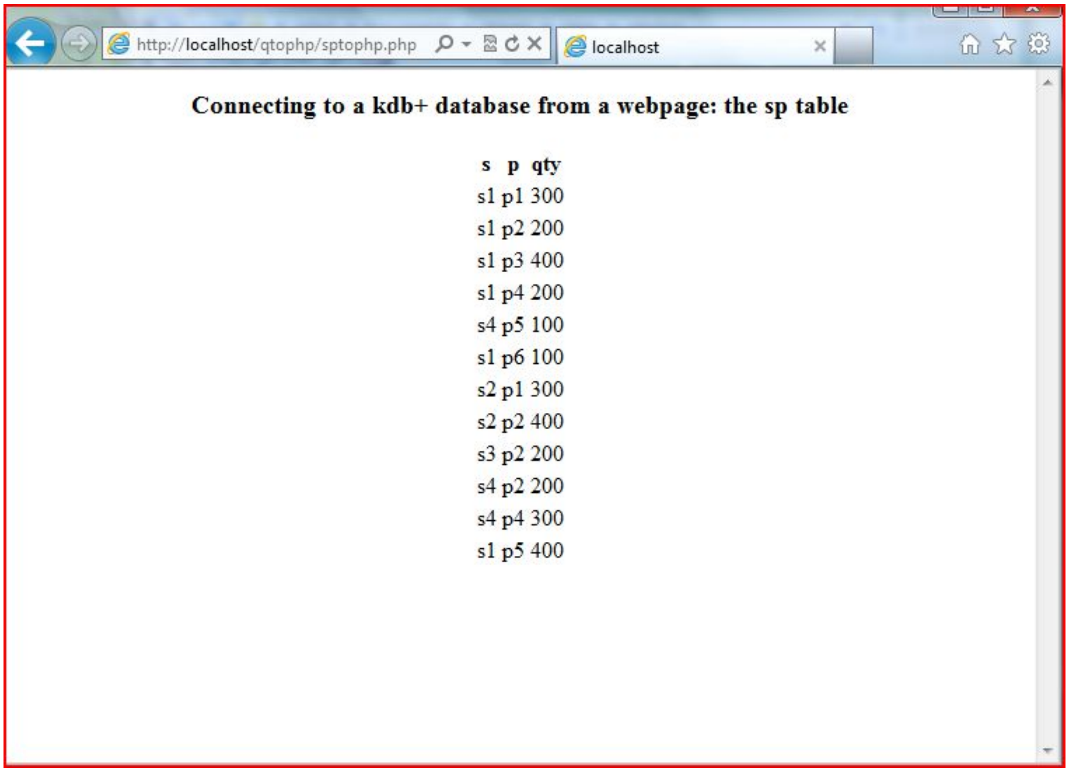
7. Open a q session, load the sp.q script and listen on port 5001:



```
KDB+ 2.7 2011.04.30 Copyright (C) 1993-2011 Kx Systems
w32/ 2()core 1982MB johnl johnl-pc 10.20.21.80 PLAY 2011.07.29

q)\l sp.q
+`p`city!(`p$`p1`p2`p3`p4`p5`p6`p1`p2;`london`london`london`london`london`lon..
(+(,`color)!,`blue`green`red)!+(,`qty)!,900 1000 1200
+`s`p`qty!(`s$`s1`s1`s1`s2`s3`s4;`p$`p1`p4`p6`p2`p2`p4;300 200 100 400 200 300)
q)sp
s  p  qty
---------
s1 p1 300
s1 p2 200
s1 p3 400
s1 p4 200
s4 p5 100
s1 p6 100
s2 p1 300
s2 p2 400
s3 p2 200
s4 p2 200
s4 p4 300
s1 p5 400
q)\p 5001
q)
```

8. Finally, open a browser and view the sp table on a webpage:

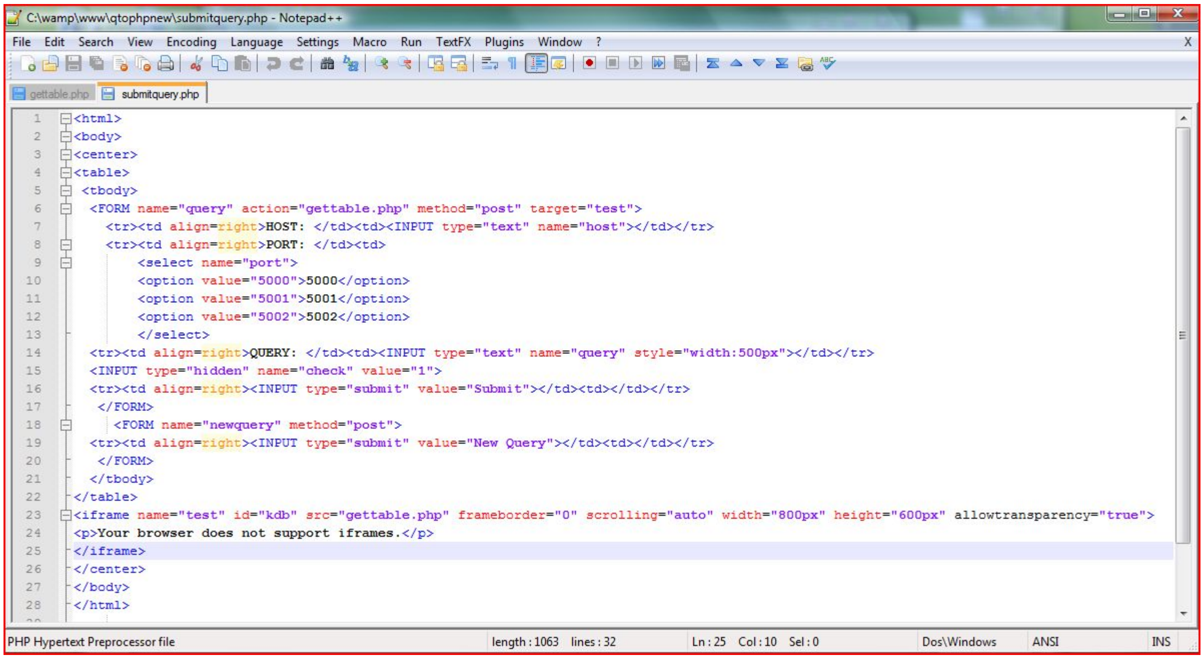


Connecting to a kdb+ database from a webpage: the sp table

s	p	qty
s1	p1	300
s1	p2	200
s1	p3	400
s1	p4	200
s4	p5	100
s1	p6	100
s2	p1	300
s2	p2	400
s3	p2	200
s4	p2	200
s4	p4	300
s1	p5	400

3 Connecting kdb+ and PHP using a GET request and an iframe

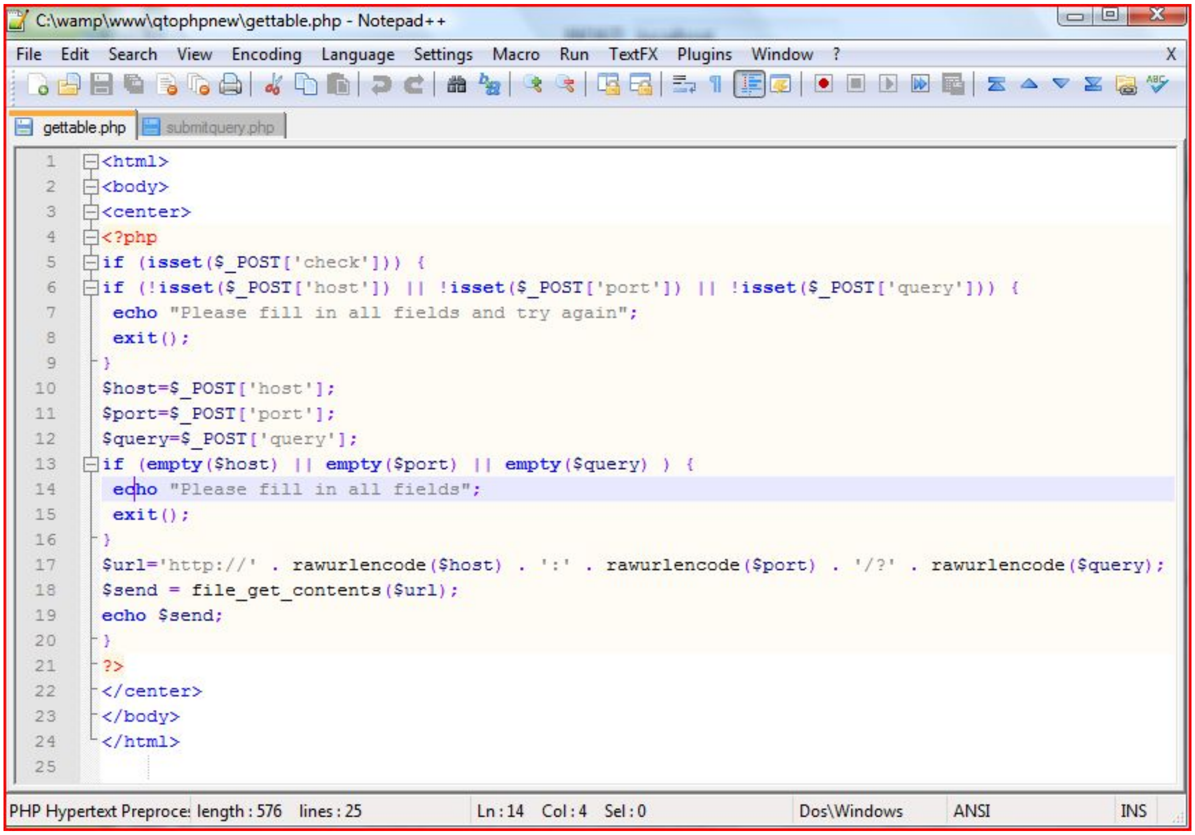
Here, we use a webform to enter a host, port and query, with the resulting query sent to the q server and returned where it is displayed using an iframe. The script submitquery.php contains the webform and iframe.


C:\wamp\www\qtophpnew\submitquery.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
gettable.php
submitquery.php

```
<html>
<body>
<center>
<table>
 <tbody>
  <FORM name="query" action="gettable.php" method="post" target="test">
    <tr><td align=right>HOST: </td><td><INPUT type="text" name="host"></td></tr>
    <tr><td align=right>PORT: </td><td>
        <select name="port">
        <option value="5000">5000</option>
        <option value="5001">5001</option>
        <option value="5002">5002</option>
        </select>
  <tr><td align=right>QUERY: </td><td><INPUT type="text" name="query" style="width:500px"></td></tr>
  <INPUT type="hidden" name="check" value="1">
  <tr><td align=right><INPUT type="submit" value="Submit"></td><td></td></tr>
   </FORM>
     <FORM name="newquery" method="post">
  <tr><td align=right><INPUT type="submit" value="New Query"></td><td></td></tr>
   </FORM>
  </tbody>
</table>
<iframe name="test" id="kdb" src="gettable.php" frameborder="0" scrolling="auto" width="800px" height="600px" allowtransparency="true">
<p>Your browser does not support iframes.</p>
</iframe>
</center>
</body>
</html>
```

PHP Hypertext Preprocessor file
length : 1063 lines : 32
Ln : 25 Col : 10 Sel : 0
Dos\Windows
ANSI
INS

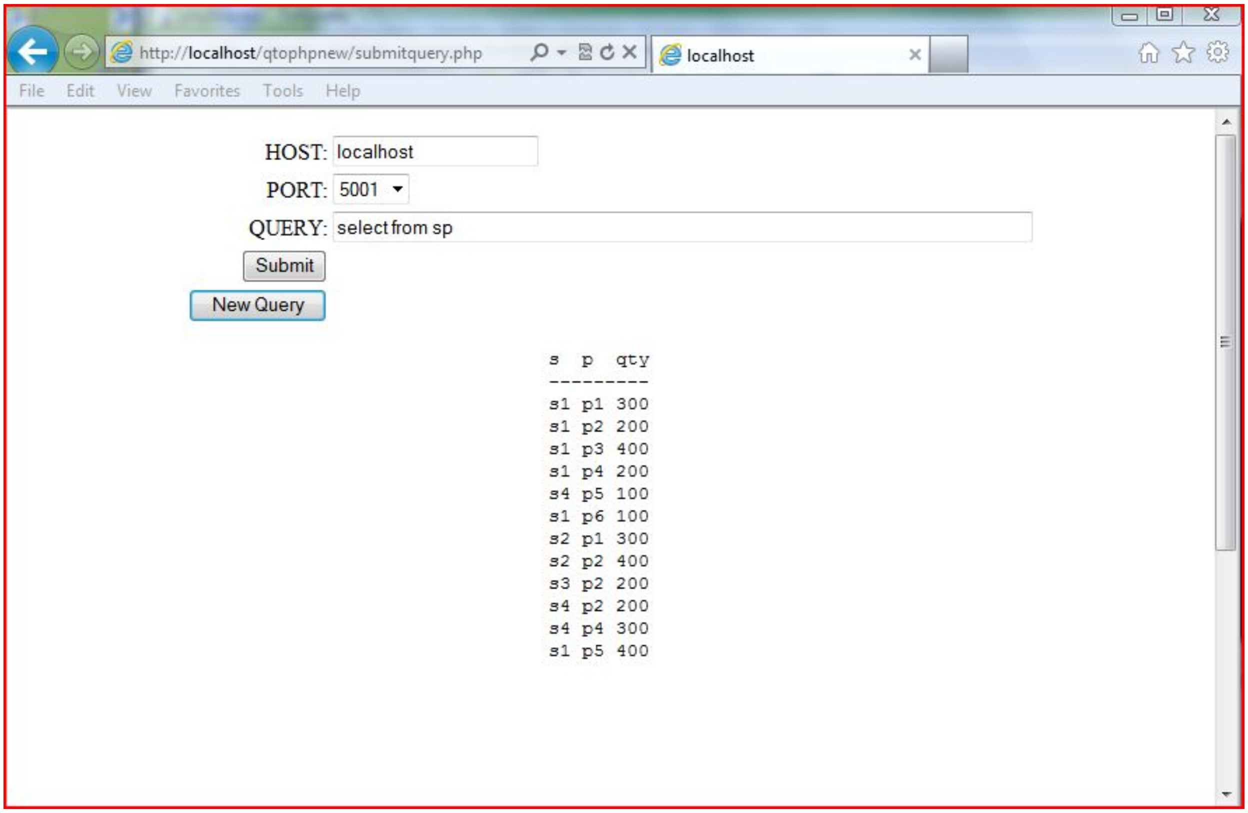
When the form is submitted the script gettable.php is used to put together the http GET request that will be sent to the q server using the php command file_get_contents($url). This request is handled by q using .z.ph and the result returned to gettable.php where it is echoed to the screen. Note that an iframe is used so that the result is returned to the same page as the webform.


C:\wamp\www\qtophpnew\gettable.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
gettable.php
submitquery.php
PHP Hypertext Preproce: length : 576 lines : 25
Ln : 14 Col : 4 Sel : 0
Dos\Windows
ANSI
INS

```
<html>
<body>
<center>
<?php
if (isset($_POST['check'])) {
if (!isset($_POST['host']) || !isset($_POST['port']) || !isset($_POST['query'])) {
 echo "Please fill in all fields and try again";
 exit();
}
$host=$_POST['host'];
$port=$_POST['port'];
$query=$_POST['query'];
if (empty($host) || empty($port) || empty($query) ) {
 echo "Please fill in all fields";
 exit();
}
$url='http://' . rawurlencode($host) . ':' . rawurlencode($port) . '/?' . rawurlencode($query);
$send = file_get_contents($url);
echo $send;
}
?>
</center>
</body>
</html>

```

As an example, open a q session, load the sp table and listen on port 5001. Now open submitquery.php, enter a port and select a port from the drop down menu (localhost and 5001 in this case). Then when you hit submit, the result is displayed in the iframe.



HOST: localhost

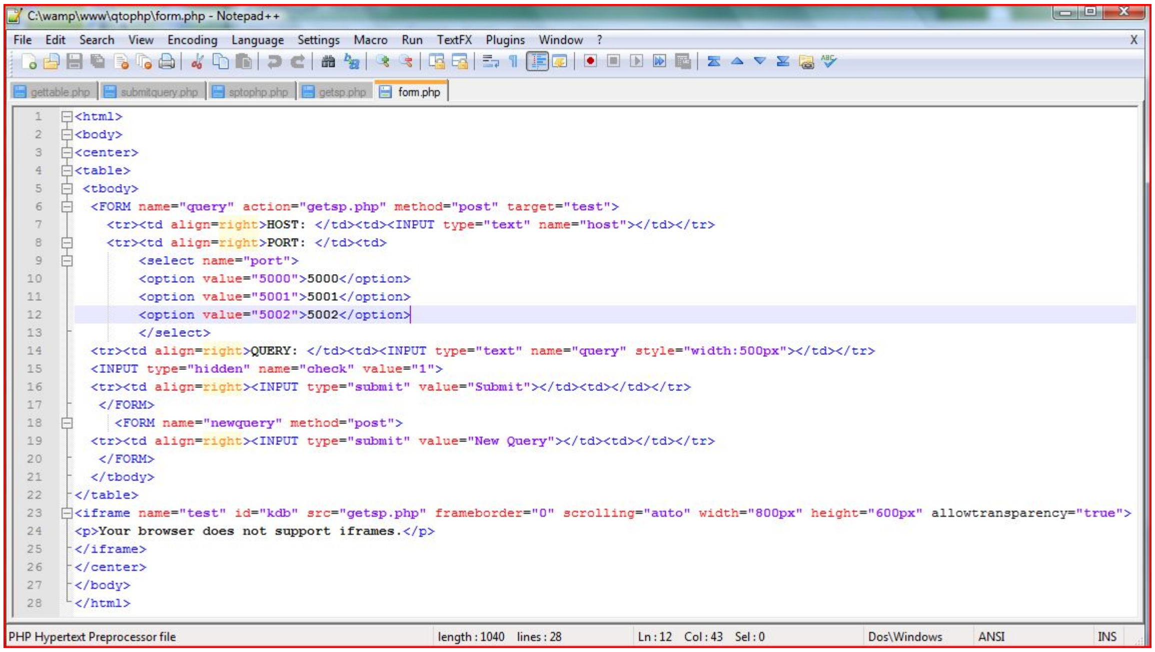
PORT: 5001

QUERY: select from sp

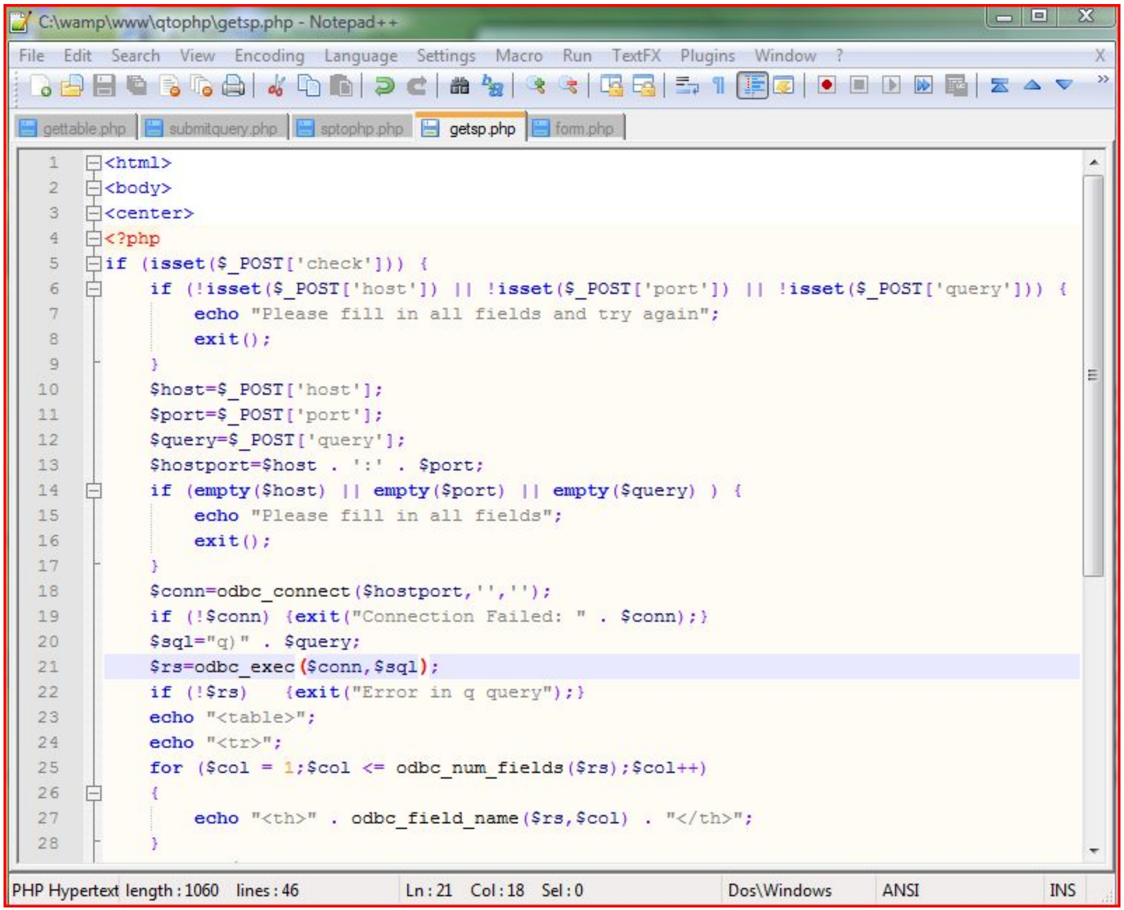
```
s  p  qty
---------
s1 p1 300
s1 p2 200
s1 p3 400
s1 p4 200
s4 p5 100
s1 p6 100
s2 p1 300
s2 p2 400
s3 p2 200
s4 p2 200
s4 p4 300
s1 p5 400
```

4 Connecting from a webpage using ODBC, a webform and an iframe

Combining the previous two sections, a webform can be used to submit a query to a q server using ODBC with the result displayed using an iframe.

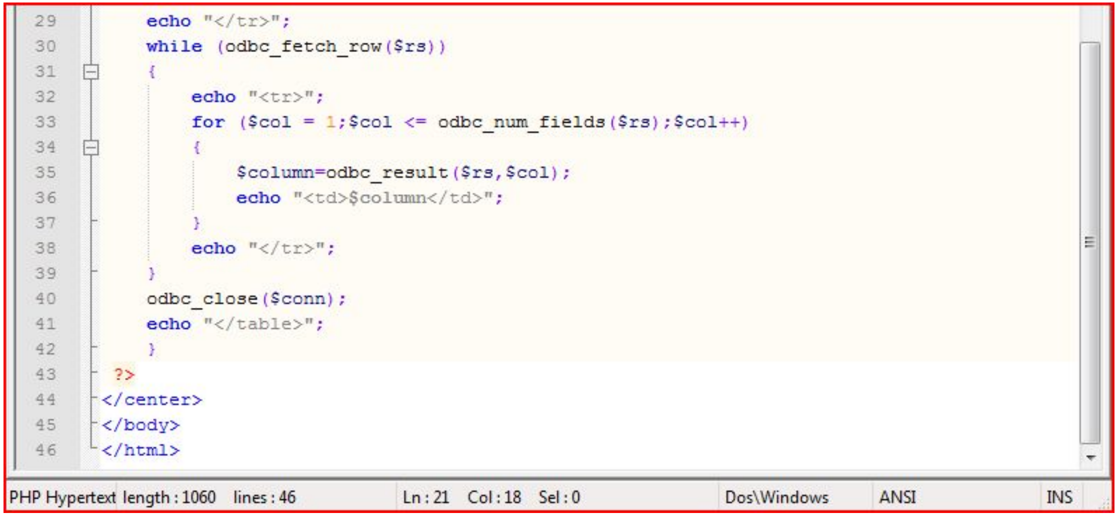

C:\wamp\www\qtophp\form.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
gettable.php
submitquery.php
sptophp.php
getsp.php
form.php
PHP Hypertext Preprocessor file
length : 1040 lines : 28
Ln : 12 Col : 43 Sel : 0
Dos\Windows
ANSI
INS

```
<html>
<body>
<center>
<table>
 <tbody>
  <FORM name="query" action="getsp.php" method="post" target="test">
    <tr><td align=right>HOST: </td><td><INPUT type="text" name="host"></td></tr>
    <tr><td align=right>PORT: </td><td>
        <select name="port">
        <option value="5000">5000</option>
        <option value="5001">5001</option>
        <option value="5002">5002</option>
        </select>
  <tr><td align=right>QUERY: </td><td><INPUT type="text" name="query" style="width:500px"></td></tr>
  <INPUT type="hidden" name="check" value="1">
  <tr><td align=right><INPUT type="submit" value="Submit"></td><td></td></tr>
   </FORM>
     <FORM name="newquery" method="post">
  <tr><td align=right><INPUT type="submit" value="New Query"></td><td></td></tr>
   </FORM>
  </tbody>
</table>
<iframe name="test" id="kdb" src="getsp.php" frameborder="0" scrolling="auto" width="800px" height="600px" allowtransparency="true">
<p>Your browser does not support iframes.</p>
</iframe>
</center>
</body>
</html>
```

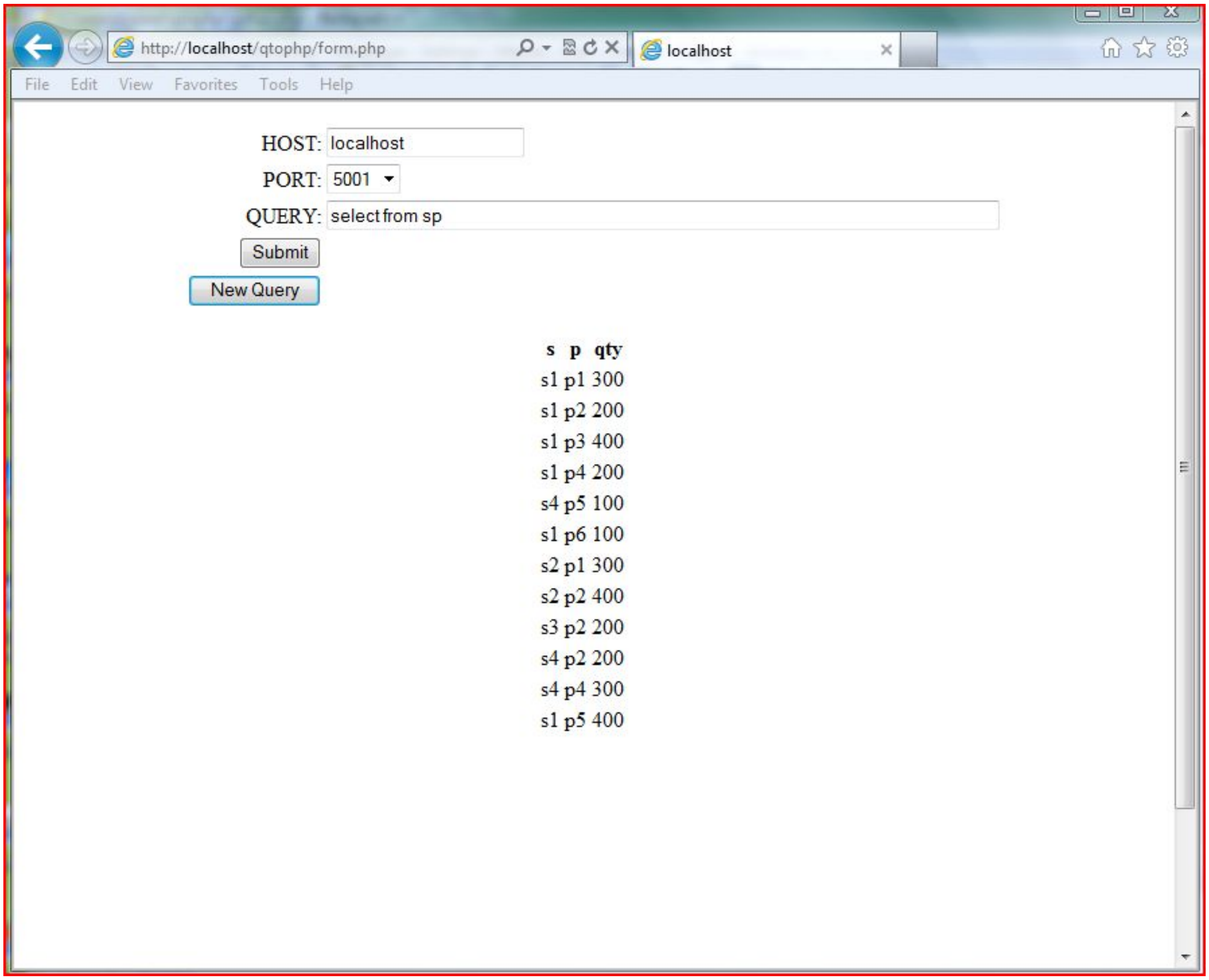


C:\wamp\www\qtophp\getsp.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins Window ?
gettable.php
submitquery.php
sptophp.php
getsp.php
form.php
PHP Hypertext length : 1060 lines : 46
Ln : 21 Col : 18 Sel : 0
Dos\Windows
ANSI
INS

```
<html>
<body>
<center>
<?php
if (isset($_POST['check'])) {
    if (!isset($_POST['host']) || !isset($_POST['port']) || !isset($_POST['query'])) {
        echo "Please fill in all fields and try again";
        exit();
    }
    $host=$_POST['host'];
    $port=$_POST['port'];
    $query=$_POST['query'];
    $hostport=$host . ':' . $port;
    if (empty($host) || empty($port) || empty($query) ) {
        echo "Please fill in all fields";
        exit();
    }
    $conn=odbc_connect($hostport,'','');
    if (!$conn) {exit("Connection Failed: " . $conn);}
    $sql="q)" . $query;
    $rs=odbc_exec($conn,$sql);
    if (!$rs)   {exit("Error in q query");}
    echo "<table>";
    echo "<tr>";
    for ($col = 1;$col <= odbc_num_fields($rs);$col++)
    {
        echo "<th>" . odbc_field_name($rs,$col) . "</th>";
    }
```



```
    echo "</tr>";
    while (odbc_fetch_row($rs))
    {
        echo "<tr>";
        for ($col = 1;$col <= odbc_num_fields($rs);$col++)
        {
            $column=odbc_result($rs,$col);
            echo "<td>$column</td>";
        }
        echo "</tr>";
    }
    odbc_close($conn);
    echo "</table>";
    }
 ?>
</center>
</body>
</html>
```


http://localhost/qtophp/form.php
localhost
File Edit View Favorites Tools Help
Submit
New Query

HOST: localhost

PORT: 5001

QUERY: select from sp

s	p	qty
s1	p1	300
s1	p2	200
s1	p3	400
s1	p4	200
s4	p5	100
s1	p6	100
s2	p1	300
s2	p2	400
s3	p2	200
s4	p2	200
s4	p4	300
s1	p5	400

5 Comments

This report offers a proof of principle on how to access kdb+ data from a webpage that is written in PHP using either the kdb+ ODBC server or a HTTP GET request. Following this approach, more complex implementations are possible, allowing users interactive access to a kdb+ server via a webpage, or the development of dynamic trading applications. For further information on how to customize the kdb+ webserver see
https://code.kx.com/trac/wiki/Cookbook/CustomWeb