# Accessing Web Services using LiveCode

This lesson gives a example on how to access a web service from LiveCode. In this example we will be creating a sample php web service script and LiveCode. Using web services can save a lot of extra work. Using them gives us an advantage to access the database from the server instead of having a temp database for Mobile Apps. All older web services you have can be called by this method. We have just showed a sample web service with a XML to show that it can be done easily. You are welcomed to try out different web services using them. You can download the associated LiveCode stack, php script and database script here

WebService_for_LiveCode.zip (http://www.livecode.com/downloads/WebService_for_LiveCode.zip)

## Step 1- Creating a web service file in php

This code outputs XML, which can be accessed from LiveCode. Here we have used a test database to retrieve data. Place this file in a web server and try running it using your own test database or you can download the test script along with the stacks. Please note before running the script you will have to create a database and the table.

```
<?php
        $number_of_posts = 10;
        $format = 'xml';
     $link = mysql_connect('localhost','root','root') or die('Cannot
connect to the DB');
          @mysql_select_db('home_food',$link) or die('Cannot select the
DB');
          $query = "SELECT name from reg_users";
           $result = @mysql_query($query,$link) or die('Errant query:
 '.$query) ;
        $posts = array();
        if(@mysql_num_rows($result)) {
                while($post = @mysql_fetch_assoc($result)) {
                        $posts[] = array('post'=>$post);
                }
        }
        if($format == 'json') {
                header('Content-type: application/json');
                echo json_encode(array('posts'=>$posts));
        }
         else {
            header('Content-type: text/xml');
            echo '<posts>';
            foreach($posts as $index => $post) {
              if(is_array($post)) {
                foreach($post as $key => $value) {
                  if(is_array($value)) {
                    foreach($value as $tag => $val) {
                      echo
'<',$tag,'>',htmlentities($val),'</',$tag,'>';
                    }
                  }
                }
              }
            }
            echo '</posts>';
          }
        @mysql_close($link);
?>
```

## Output of Web Service

(/m/4070/l/40932-accessing-web-services-using-livecode/show_image?image_id=1171737)

<posts>

<name>karthik</name>

<name>Thomas</name>

<name>Ravi</name>

<name>Alex</name>

<name>Kirushanth</name>

</posts>

## LiveCode Stack to access Web Service

```
LiveCode Stack to access Web Service
```

The stack contains a Button and a Scrolling Text Field

## Tell the button to load data from the web service

```
Tell the button to load data from the web service
```

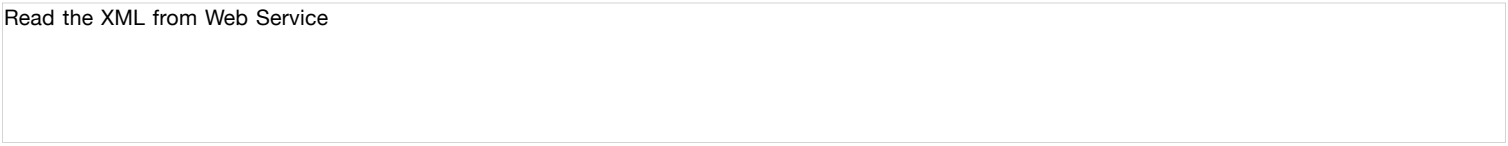
(/m/4070/l/40932-accessing-web-services-using-livecode/show_image?image_id=1171734)

Begin the script with the following code

```
on mouseUp
# When the button is clicked, load up the preferences and put them into
the field
   loadPreferences
end mouseUp

command loadPreferences
# There are two parts to loading the XML Data. The first part is
reading the XML Data into memory and
# creating an XML "tree". The second part is to process the tree and
extract the wanted data from it.
# This function reads the XML Data, and returns the tree. The tree is
represented as a number, the actual
# tree structure and data is managed by LiveCode and so we don't need
to worry about it.
   local tTree
   put readPreferencesToXMLTree() into tTree
   if tTree is empty then
      exit loadPreferences
   end if

# This command reads the preferences we require from the tree and
displays them.
   processPreferencesTree tTree

# Close the XML tree. This will free up the memory that the tree was
using and prevent our
# application using more memory than it needs or "leaking" memory by
creating multiple trees
# without closing any of them.
   revDeleteXMLTree tTree
end loadPreferences
```

Note that this code doesn't do anything just yet, because we haven't yet implemented the function readPreferencesToXMLTree and the command processPreferencesTree.

## Read the XML from Web Service



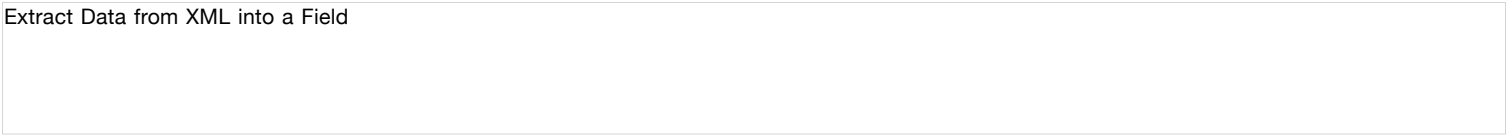

(/m/4070/l/40932-accessing-web-services-using-livecode/show_image?image_id=1171739)

Next, we implement a function to read the XML. This is done in two steps, first the XML is read into a variable like any other text file would be, secondly, an XML "tree" is created from the file. This tree allows us to manipulate the XML data easily.

The code to read the XML file and create the tree looks like this:

```
# This function reads the XML from Web Service, and turns it into an
XML Tree. The tree is then returned
# for the second part of the process.
private function readPreferencesToXMLTree
# Load the Web Services URL to a variable.
   set the itemDelimiter to slash
   local tPreferencesFile
   put URL "http://127.0.0.1:8081/livecode/samplexml1.php" into
tPreferencesFile

   local tPreferencesData, tResult
   put tPreferencesFile into tPreferencesData
   put the result into tResult

   if tResult is not empty then
      answer error "Failed to read preferences file at location: " &
tPreferencesFile
      return empty
   end if

  # Create the XML "tree" from the data, checking to make sure that the
file has loaded properly.
  # The revCreateXMLTree function will return a number (the tree's
"handle" or "id") if it succeeds,
  # otherwise it will return a message saying why it failed.
   local tTree
   put revCreateXMLTree(tPreferencesData, false, true, false) into tTree

   if tTree is not an integer then
      answer error "Failed to process preferences file with error: " &
tTree
      return empty
   end if


   return tTree
end readPreferencesToXMLTree
```

## Extract Data from XML into a Field



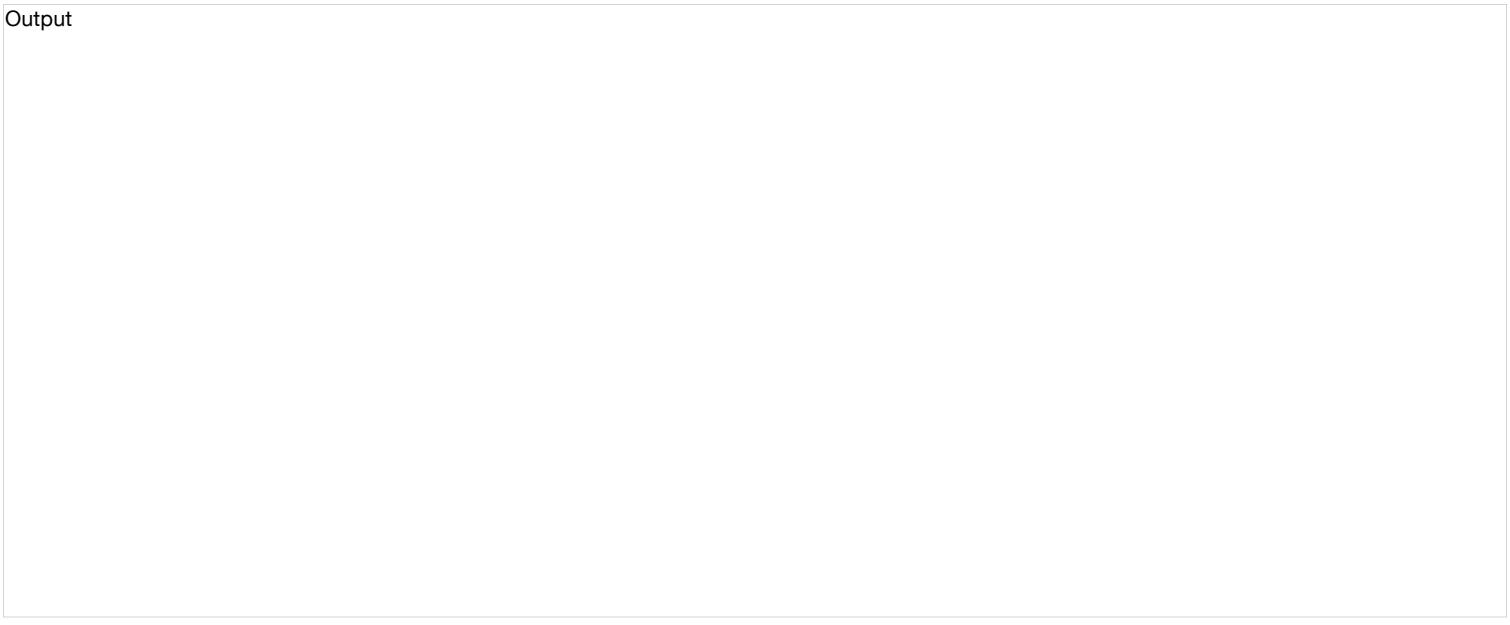

(/m/4070/l/40932-accessing-web-services-using-livecode/show_image?image_id=1171732)

Once we have the XML tree, the final step is to use LiveCode's XML library to get the required information out of it. We use a series of calls to the XML library to extract each piece of information from the tree.

```
private command processPreferencesTree pTree

        # First, we get a list of Names, then we can loop through them
and get each one in    turn.
        # The revXMLChildNames function is useful for returning a list
of nodes like this.    The last parameter is important
        # as it tells the function to return a unique specifier for each
node, allowing us      to access them correctly. This will
        # look something like:
        # name[1]
        # name[2]
        # name[3]
        local tPosts
   put revXMLChildNames(pTree, "posts", return, "name", true) into
tPosts
   local tListOfNames

        # To get each Name, we just use revXMLNodeContents again.
However here we concatentate the name of each node
   repeat for each line tName in tPosts
      put revXMLNodeContents(pTree, "posts/" & tName) & return after
tListOfNames
   end repeat

   delete the last char of tListOfNames

   local tOutput
        # Now put the List of Names retrieved into the field

   put tListOfNames after tOutput
   set the text of field "NameList" to tOutput

end processPreferencesTree
```

## Output

Output



By clicking on the button From Services, we can get the Data into our field with the help of Web Services.

## 6 Comments

### terii Tuesday Dec 05 2017 at 11:09 PM

Hi, This doesnt work on an https server : how do one configure this line : put the URL"http://127.0.0.1:8081/livecode/samplexml1.php (http://127.0.0.1:8081/livecode/samplexml1.php)" into tPreferencesFile - Does the real server name replaces 127.0.0.1 (localhost) ? what about ssl ? tks

### Elanor Buchanan Thursday Dec 07 2017 at 05:59 AM

Hi Terii

There was an error in the lesson code, which I have now updated. The line should not have a 'the' in it it should be

put URL "http://127.0.0.1:8081/livecode/samplexml1.php (http://127.0.0.1:8081/livecode/samplexml1.php)" into tPreferencesFile

You can replace the localhost with your real server name.

I hope that helps, thanks for bringing this to our attention.

Kind regards

Elanor

### Kim Monday Mar 19 2018 at 04:17 PM

Hi. Could you point me to a lesson / example of the reverse process - how I would use LC to write XML an document to a web service? Regards. K

### Heather Laine Tuesday Mar 20 2018 at 06:36 AM

I'm not sure we have such a lesson at present, I can request one. You could look at this one which might be some help:http://lessons.livecode.com/m/4071/l/7011-how-to-read-in-data-from-an-xml-file (http://lessons.livecode.com/m/4071/l/7011-how-to-read-in-data-from-an-xml-file)

### Akashdeep Tuesday Jul 17 2018 at 04:17 PM

If you get "Call to undefined function mysql_connect().." error then you can use the following PHP script

Replace 'Username','Password','NameOfDatabase' with your respective credentials

```
$query = "SELECT name from users";

$result=$link->query($query);

$posts = array();
if(@mysqli_num_rows($result)) {
while($post = @mysqli_fetch_assoc($result)) {
$posts[] = array('post'=>$post);

}
}

if($format == 'json') {
header('Content-type: application/json');
echo json_encode(array('posts'=>$posts));
}
else {
```

```
header('Content-type: text/xml');
echo '';
foreach($posts as $index => $post) {
if(is_array($post)) {
foreach($post as $key => $value) {
if(is_array($value)) {
foreach($value as $tag => $val) {
echo '',htmlentities($val),'';
}
}
}
}
}
echo '';
}
```

---

## Akashdeep Tuesday Jul 17 2018 at 04:21 PM

Correction:

If you get "Call to undefined function mysql_connect().." error then you can use the following PHP script

Replace 'Username','Password','NameOfDatabase' with your respective credentials

```
$link = mysqli_connect('localhost','root','','livecode') or die('Cannot connect to the DB');
$query = "SELECT name from users";

$result=$link->query($query);

$posts = array();
if(@mysqli_num_rows($result)) {
while($post = @mysqli_fetch_assoc($result)) {
$posts[] = array('post'=>$post);

}
}

if($format == 'json') {
header('Content-type: application/json');
echo json_encode(array('posts'=>$posts));
}
else {
header('Content-type: text/xml');
echo '';
foreach($posts as $index => $post) {
if(is_array($post)) {
foreach($post as $key => $value) {
if(is_array($value)) {
foreach($value as $tag => $val) {
echo '',htmlentities($val),'';
}
}
}
}
}
echo '';
}

@mysqli_close($link);

?>
```