AuditLog

Purpose

Document all changes to tables and keep an auditlog with user who did those, timestamp, current and previous value.

Information

I have tried to make the installation as easy as possible.
However, with the certain knowledge that no matter how idiot proof you make something - nature will provide a better idiot, I feel I must add the following:
Despite the fact that I have used this installation procedure on 11 different applications built with AppGini, I in NO way take ANY responsibility for mistakes in the code or that YOU might make - ALWAYS backup you files before attempting a major modification!

Versions

Adjustments by Olaf Nöhring, 2021-01 (https://datenbank-projekt.de):
- removed bug when record is deleted (v1.71)
- added possibility to easily define the name of your auditor table in the beginning of the the files auditLog_functions.php and /admin/auditLog.php
- changed code in the way that now the order of fields in the database must not match the order of fields in your AppGini application anymore. In previous versions the order must be the same, otherwise it would mess up the logging. Now this problem should be solved.
- added a new function `Audit_Manually` which allows checking for changes on another table and documenting those (see description below for more information).
- transformed docs to PDF for easier editing
- changed audit_tableSQL.sql to make larger fields for table and fieldnames

Adjustments by Olaf Nöhring, 2019-12 (https://datenbank-projekt.de):
- improved INSERTION: Now, all non-empty fields are written to the auditor table after insert. Until now, only the new primary key was written. - you can easily use a different table name for teh auditor. Simply adjust
$audittablename = "auditor";
in function table_after_change in auditLog_functions.php (and the setup sql or course). - changed auditor table name from Auditor to auditor (script and setup). Note: On Linux systems the tablenames are case sensitive!

Adjustments by Olaf Nöhring, 2019-06 (https://datenbank-projekt.de):
- Trick: Added in docs. Remove access to Auditor from Admin menu, but use regular AppGini table instead, so Auditor stays even when application is regenerated.
- Trick: Remove changes from 'application_root/lib.php', instead place code in config.php which stays in place, even when the application is regenerated.
- Changes to auditLog_functions.php, added , $eo to SQL queries (following vaalonv tip)
- Instead of foreign keys you will see the values the user actually selected (old code to see FKs still

in the file). For this to work correct, make sure the order of the fields in your database is exactly like the order of the fields in specific table in AppGini!

Installation

ATTENTION!!
Note 1. For the Search/Replace I used 'Notepad++' available here : Notepad++ Home Page

Note 2. I suggest that you wait till your application is ready to go to production before making these changes - although this is NOT essential - (with the proviso that you BACK UP YOUR FILES FIRST!)

Note 3. When it comes to the tedious task of doing the Search/Replace in the Hooks folder. I suggest that you copy the hook files ONLY for the tables you wish to monitor into a separate directory and then BACKUP that directory. I further suggest that you not include 'Lookup' tables. This way, you can do it speedily using Notepad++'s Search/Replace facility 'Find in Files' and do them all in just six shots.

---

1. Create the Audit Log Table using the *audit_tableSQL.sql* file provided.

2. Copy and then Extract the *auditlog_files.zip* in your application root folder.

The zip file contain only 2 files:
'auditLog_functions.php' -> The functions that allow the audit log to work.
'admin/auditLog.php' -> A table (filterable and pageable) that will be added to the Admin Menu Options.

3. Essential File Modifications

3.1 In 'application_root/config.php':

Add the following to the bottom of the file.

```
if (session_status() == PHP_SESSION_NONE) { session_start(); }
$_SESSION['dbase'] = $dbDatabase;
if (!function_exists('table_before_change')) {
      $currDir = dirname(__FILE__);
      @require("$currDir/hooks/auditLog_functions.php");
}
```

3.2 In 'admin/incHeader.php' - (Adds page to the Admin Menu Options.)

> Trick:
> If you create a table in AppGini with the name as the Auditor table (i.e. Auditor) and the same fields (case sensitive) as the in the Auditor table, you can build a regular Audit-Table button from AppGini and let user access that with the regular permissions.
> Maybe you want to make sure, that noone can change anything in that table.

If you to this, you do not need to make changes in 'admin/incHeader.php' as described now (and thus Auditor will stay in your application even when regenerated.

Do find for:

```
<a class="navbar-brand" href="pageHome.php"><span class="text-
warning"><i class="glyphicon glyphicon-wrench"></i> Admin
Area</span></a>
```

and replace with:

```
<a class="navbar-brand" href="pageHome.php"><span class="text-
warning"><i class="glyphicon glyphicon-wrench"></i> Admin
Area</span></a><a class="navbar-brand" href="auditLog.php"><span
class="text-warning-1"><i class="glyphicon glyphicon-tasks"></i> Audit
Log</span></a>
```

4. The HOOKs folder. After you've read Note 3, above:

In the temp folder that contains the Hooks for the tables that you wish to monitor:
A. Do 'find in files' for: (Remember to set the correct directory!)

```
init(&$options, $memberInfo, &$args){
```

and replace with:

```
init(&$options, $memberInfo, &$args){
$_SESSION ['tablenam'] = $options→TableName;
$_SESSION ['tableID'] = $options→PrimaryKey;
$tableID = $_SESSION ['tableID'];
```

B. Do 'find in files' for:

```
after_insert($data, $memberInfo, &$args){
```

and replace with:

```
after_insert($data, $memberInfo, &$args){
table_after_change($_SESSION ['dbase'], $_SESSION['tablenam'],
$memberInfo['username'], $memberInfo['IP'], $data['selectedID'],
$_SESSION['tableID'], "INSERTION");
```

C. Do 'find in files' for:

```
before_update(&$data, $memberInfo, &$args){
```

and replace with:

```
before_update(&$data, $memberInfo, &$args){
table_before_change($_SESSION['tablenam'], $data['selectedID'],
$_SESSION['tableID']);
```

D. Do 'find in files' for:

```
after_update($data, $memberInfo, &$args){
```

and replace with:

```
after_update($data, $memberInfo, &$args){
table_after_change($_SESSION ['dbase'], $_SESSION['tablenam'],
$memberInfo['username'], $memberInfo['IP'], $data['selectedID'],
$_SESSION['tableID'], "UPDATE");
```

E. Do 'find in files' for:

```
before_delete($selectedID, &$skipChecks, $memberInfo, &$args){
```

and replace with:

```
before_delete($selectedID, &$skipChecks, $memberInfo, &$args){
table_before_change($_SESSION['tablenam'], $selectedID,
$_SESSION['tableID']);
```

F. Do 'find in files' for:

```
after_delete($selectedID, $memberInfo, &$args){
```

and replace with:

```
after_delete($selectedID, $memberInfo, &$args){
table_after_change($_SESSION ['dbase'], $_SESSION['tablenam'],
$memberInfo['username'], $memberInfo['IP'], $selectedID,
$_SESSION['tableID'], "DELETION");
```

Remember to copy the files from the temp directory created in Note 3 back to the original Hooks folder!

AuditLog should now be working! Using this technique will allow you to keep other modifications made to the Hooks folder.

Documenting changes for another table, Audit_Manually

In version 1.7 of this script a new function was introduced: `Audit_Manually`

The function allows to manual add to Auditor-table (and/or save old value to session variable). This is useful, in case you have table TA and table TB. TB is a child of TA (holds the foreign key of TA). The user does something in table TB. Following this change, you (your application) do some

action to the parent(!) record in TA. If you want to add this action (done to a record in TA) also to the audit log, you use this function.
Example setting - you need to place the code in the correct functions! (maybe before_insert, after_insert, before_delete, after_delete):
TA has primary key field pkA. User is working in TB with parent record pkA=5 ($valueOfpkA)
Now the field TA.LogEdit ($fieldName) should be changed, once something in table TB happens

Yours, Olaf & primitive_man
Discussions about this: Here