

AuditLog

Contents

Purpose.....	1
Disclaimer.....	1
Installation as Plugin-Version (recommended).....	2
Access to Audit Log.....	2
Possible Adjustments.....	3
Custom table name for the auditor table.....	3
Standard Audit Log Table.....	4
Advanced Audit Log Table (recommended).....	4
Exact position of audit call functions.....	6
Auditor Timeline.....	6
Make the Auditor Timeline accessible to your users.....	9
Extras.....	10
Documenting changes for another table: fct Audit_Manually.....	10
Manual Installation (not recommended!).....	10
Attention.....	10
Note 1.....	10
Note 2.....	11
Note 3.....	11
Step 1. Extract the <i>auditlog_files.zip</i> and copy files.....	11
Step 2. Create the Audit Log Table using the <i>audit_tableSQL.sql</i> file provided.....	11
Step 3. Essential File Modifications.....	11
3.1 Include audit-base files: 'application_root/config.php'.....	11
3.2 Add page to the Admin Menu Options: 'admin/incHeader.php'.....	12
Step 4. Essential /hooks/folder-files modification.....	12
AppGini Code for Auditor table.....	14
History / Versions / Changes.....	14

Purpose

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

Discussions about this: <https://forums.appgini.com/phpbb/viewtopic.php?f=4&t=1369>

Disclaimer

We have tried to make the installation as easy as possible. [Landinialejandro](#) provided a plugin that makes installation even easier – see below. However, with the certain knowledge that no matter how idiot proof you make something, nature will provide a better idiot, we feel we must add the

following:

Despite the fact that this extension was used in several different applications, built with AppGini, we in NO way take ANY responsibility for mistakes in the code or that YOU might make. ALWAYS backup you files and your database before attempting a major modification!

Installation as Plugin-Version (**recommended**)

This works only if you have acquired some other official AppGini plugin. If you do not have an official plugin or you want to use the manual installation instead, please read [below: Manual Installation \(not recommended!\)](#).

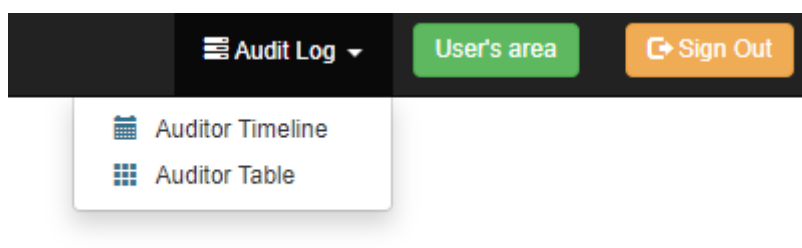
New from version 1.71: The AppGini forum member [landinialejandro](#) created a plugin which makes installation much easier. Please see the forum (<https://forums.appgini.com/phpbb/viewtopic.php?f=4&t=1369>) and/or the github page where he placed it (https://github.com/myappgini/sbm_audit_log)

If you decide to use this plugin for installation (**yes, it's the recommended way**),

1. download the latest plugin from it's homepage https://github.com/myappgini/sbm_audit_log
2. read the README.md of the plugin.
3. If needed/wished, update the folder app-resources of the plugin by putting all files from this ZIP directly into that folder
4. upload the plugin to your AppGini generated site as usual
5. go to the admin area of your site and install the plugin as usual.

Access to Audit Log

Once installed (plugin, or manual with auditor-table-link in the admin area) you as superadmin have the opportunity to see all changes in the application. You will find a new menu in the admin area which should look like this:



Audit Log menu in the Admin Area

Possible Adjustments

Custom table name for the auditor table

If you prefer, you can easily change the name of the auditor table for example to match some table-prefix you are using. This should be done *before* running the installation via plugin as some file contents needs to be changed.

Note: table names are case-sensitive (at least in Linux). Therefor I suggest lowercase table names.

Open `audit_tableSQL.sql`

Search for

```
`auditor` (
```

replace with

```
`your_tablename_here` (
```

Open `/hooks/auditLog_functions.php`

Search for

```
define("AUDITTABLENAME", 'auditor');
```

replace with

```
define("AUDITTABLENAME", 'your_tablename_here');
```

Open `/hooks/auditLog_timeline.php`

Search for

```
define("AUDITTABLENAME", 'auditor');
```

replace with

```
define("AUDITTABLENAME", 'your_tablename_here');
```

Open `/admin/auditLog.php`

Search for

```
define("AUDITTABLENAME", 'auditor');
```

replace with

```
define("AUDITTABLENAME", 'your_tablename_here');
```

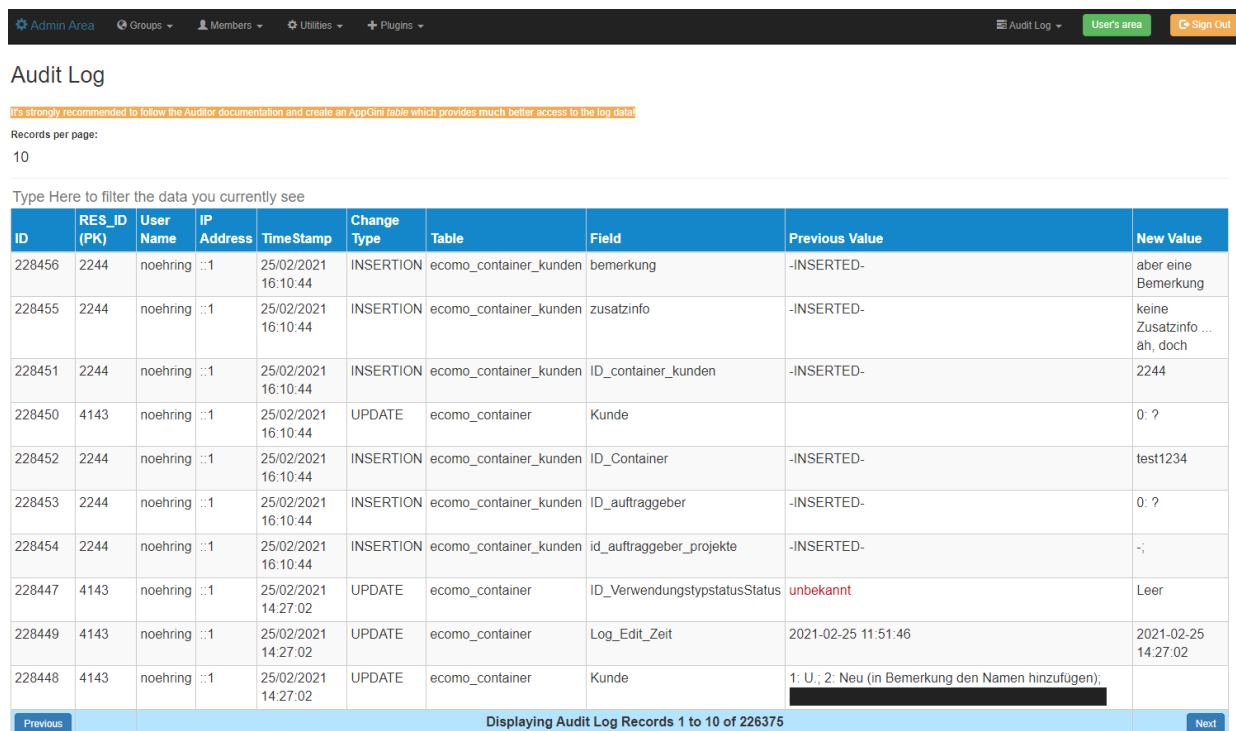
Now run the installation using the plugin.

Standard Audit Log Table

The *Auditor Table* view provides all information, but in a *very basic* form. If you want to have easier access and better filters, please read [below: Advanced Audit Log Table \(recommended\)](#):

You can set the number of records you want to see at once. Just press the enter key after you set the desired number and the page will reload.

Note: The simple search function searches only through the records that are displayed on the page. It does not search the complete auditor log table.



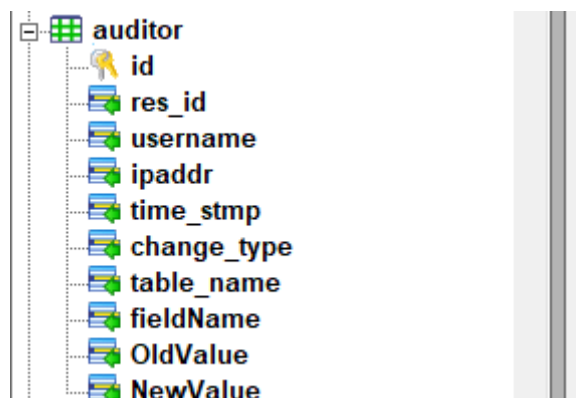
ID	RES_ID (PK)	User Name	IP Address	TimeStamp	Change Type	Table	Field	Previous Value	New Value
228456	2244	noehring	::1	25/02/2021 16:10:44	INSERTION	ecom_container_kunden	bemerkung	-INSERTED-	aber eine Bemerkung
228455	2244	noehring	::1	25/02/2021 16:10:44	INSERTION	ecom_container_kunden	zusatzinfo	-INSERTED-	keine Zusatzinfo ... äh, doch
228451	2244	noehring	::1	25/02/2021 16:10:44	INSERTION	ecom_container_kunden	ID_container_kunden	-INSERTED-	2244
228450	4143	noehring	::1	25/02/2021 16:10:44	UPDATE	ecom_container	Kunde		0: ?
228452	2244	noehring	::1	25/02/2021 16:10:44	INSERTION	ecom_container_kunden	ID_Container	-INSERTED-	test1234
228453	2244	noehring	::1	25/02/2021 16:10:44	INSERTION	ecom_container_kunden	ID_auftraggeber	-INSERTED-	0: ?
228454	2244	noehring	::1	25/02/2021 16:10:44	INSERTION	ecom_container_kunden	id_auftraggeber_projekte	-INSERTED-	-;
228447	4143	noehring	::1	25/02/2021 14:27:02	UPDATE	ecom_container	ID_VerwendungstypstatusStatus	unbekannt	Leer
228449	4143	noehring	::1	25/02/2021 14:27:02	UPDATE	ecom_container	Log_Edit_Zeit	2021-02-25 11:51:46	2021-02-25 14:27:02
228448	4143	noehring	::1	25/02/2021 14:27:02	UPDATE	ecom_container	Kunde	1: U.; 2: Neu (in Bemerkung den Namen hinzufügen);	

Previous Displaying Audit Log Records 1 to 10 of 226375 Next

Standard Audit Log in the admin area. Please note, that the IP address is correct. The screenshot was done in a local testing environment, this it's the IPv6 format of the localhost address (IPv4: 127.0.0.1).

Advanced Audit Log Table (recommended)

You can have a much better display and search capabilities when you add an extra table to your AppGini application. Just create a simple table in your AppGini project and make sure it has the same name and the same fields as the auditor table in the database!



Auditor table in your AppGini project.

You can even copy the code for this AppGini table directly from this documentation into the clipboard and then paste it as a new table in AppGini. You find the code [below](#): [AppGini Code for Auditor table](#)

Of course, you can label the fields as you like (in your own language)). When following this, AppGini will create a standard page for the table which you can search and filter with the standard AppGini functions which are far more advanced than the solution above.

Your table could look like this:

Table title set in AppGini

Auditor Basis Daten

ID	res_id	username	ipaddr	time_stmp	change_type	table_name	fieldName	OldValue	NewValue
ID	Primärschlüssel	Username	IP Adresse	Zeitstempel	Veränderungstyp	Tabelle	Feld	Alter Wert	Neuer Wert
186294	1740	...	::1	2021-01-21 13:57:54	UPDATE	...	Log_LetzerBearbeiter
186293	1740	...	::1	2021-01-21 13:57:54	UPDATE	...	Log_Edit_Zeit	2020-11-09 12:04:56	2021-01-21 13:57:54
186292	1740	...	::1	2021-01-21 13:57:54	UPDATE	...	Bemerkung
186291	4	...	::1	2021-01-20 18:00:46	INSERTION	...	Bestellnr	-INSERTED-	...
186290	4	...	::1	2021-01-20 18:00:46	INSERTION	...	StatusGeeandert	-INSERTED-	2021-01-20 18:00:46
186289	4	...	::1	2021-01-20 18:00:46	INSERTION	...	ID_bestellung_status	-INSERTED-	Neu
186288	4	...	::1	2021-01-20 18:00:46	INSERTION	...	BestellungAktiv	-INSERTED-	Nein
186287	4	...	::1	2021-01-20 18:00:46	INSERTION	...	StatusUpdates	-INSERTED-	Ja
186286	4	...	::1	2021-01-20 18:00:46	INSERTION	...	zieldatum	-INSERTED-	21.01.2021
186285	4	...	::1	2021-01-20 18:00:46	INSERTION	...	bestelldatum	-INSERTED-	20.01.2021

Auditor-Table created in the AppGini project and here in the generated application. Red descriptions added to show which labels are the columns from the auditor table.

You can – and should – **set permissions** to that Auditor table:

View permission is enough! It's suggested, that you **do not allow** anyone to insert, edit, update or delete in that table.

Exact position of audit call functions

The plugin-installation-script (and the [Manual Installation \(not recommended!\)](#) instructions [below](#)) place the call for the audit log functions (`table_before_change` and `table_after_change`) directly after the opening of certain functions in the `hooks/tablename.php` file. You might have the need to adjust the position of the calls of the auditor functions.

For example, you may need to change some data directly after the user clicked the save-changes-button, this in the `before_update` function – or – that you need to change data after it has been saved. In these cases you need to move the call for the auditor function by hand to a position after your changes.

If you do this, everything should work. Once you run the installation plugin again, the plugin should notice that the call already exist (does not matter where in the function), notify you about the existence during installation, skip the function but continue with the next. If the plugin has a problem it will notify you that you have to do some manual labor.

Auditor Timeline

The *Auditor Timeline* was added in version 1.9 to this AppGini extension. It allows you to get the state of each record in each table (if all tables use the auditor script) at every change. This means, you can easily see and compare before and after of any record at a given timestamp.

The *Auditor Timeline* provides a wizard which leads you through the process. The steps are as follows:

Step 1: Choose a table from all tables that can be found in the Audit Log.

Step 2: Choose a record (from column `res_id`). The column `res_id` shows the primary key from the table you have chosen in step 1. The number that follows the primary key is the amount of timestamps for which changes have been recorded in the auditor for that specific record. Example: 4143 (100) would mean, the record with primary key 4143 has been logged at 100 different times (This is not equal the number of changes, as in one edit more than one field can be changed, resulting in two entries in the auditor, but both carrying the same timestamp.)

From this step on, the field to select the table is locked.

Step 3: Verify you have chosen the correct record. For this the current record is taken from the chosen table and displayed. Lookup fields are showing the values that are actually saved in the database, not those, you would see in the generated AppGini application.

If you have chosen the wrong record, simply press the *Start Over* button at any time.

From this step on, the field to select the record (primary key) is locked.

Step 4: Choose the timeframe for which the timeline should be generated. The wizard will suggest the first timestamp as start and the last as end for the timeline. Keep the date/time-format or you will break the timeline (no checks for dates done in the background).

This last step shows some additional options.

Remove HTML: This will use the PHP function `strip_tags` before the data is displayed and remove all HTML. This might be helpful if the data contains some formatting HTML for example. Default is unchecked.

Include latest data before start timestamp (should prevent no-data-cells): If you check this (default is checked), the timeline will try to pull data for each of the columns from older auditor entries before the start that is set. This should give you a better idea what the record looked like at a certain point in time.

The next picture shows the last step of the wizard, before the timeline will be generated.

Timeline

Preparation, step 4 of 4

Select table :

ecomo_container

(only tables found in ecomodo_auditor are shown)

Select record (primary key) from table 'ecomo_container'. Number in () is number of different timestamps of changes in the auditor table for the record.

4143 (100)

(order of primary keys list: pk with newest change first)

Current state of selected record

ID_ContainerNum	ContainerCode	ID_Location	ID_Verwendungstypstatus	Status	ID_Projekte
4143	test1234	6236	2		1

Set starting date/time (YYYY-MM-DD HH:MM:SS):

2019-12-27 16:15:36

Set end date/time (YYYY-MM-DD HH:MM:SS):

2021-02-25 16:10:44

Remove HTML ☐

Include latest data before start timestamp (should prevent no-data-cells) ☒

✓ build timeline

✗ Start over

Timeline, Step 4

Once the last step is done and you decided for your options, press the *build timeline*-button.

The result could look like this:

The columns with the [] around their name are automatically generated (there are some more on the far right side).

The green fields have been changed at the time that can be seen on the left (column Timestamp). If something was not changed, the previous value will be repeated (gray text) to make it easier to grasp the state of the record (#2 for example ID_ContainerNum was not changed, but ID_Location was).

You can also see the last column in the picture (ID_ProjekteZusatz). The dashed line shows, that there is no data for that field at that timestamp – not even old data that should have been pulled. One reason could be, the field did not exist at the time of the change. Another reason might be, that changes in this table have not been documented from the very beginning.

You can hover all cells and get more information.

Timeline

PK: **4143** from **ecommo_container** between **2019-12-27 16:15:36** and **2021-02-25 16:10:44**

Columns in square brackets [and] are not from the table itself, but extra information for the timeline, automatically generated.

Cells with **this** style have changed at the given timestamp. Cells with this style have show the value the field had at the changed of other fields at the given timestamp. Cells with this style demonstrate that no data is available for the field at the given timestamp.

Hover over values for more information.

[#]	[Timestamp]	ID_ContainerNum	ContainerCode	ID_Location	ID_Verwendungstypstatus	Status	ID_Projekte	ID_ProjekteZusatz	K
1	2019-12-27 16:15:36	4143	test1234	8°F1-0/0	Leer	-			
2	2019-12-27 16:16:00	4143	test1234	8°F2-1/1	Leer	-			
3	2019-12-27 16:16:11	4143	test1234	8°F3-1/1	Leer	-			

Timeline

Make the *Auditor Timeline* accessible to your users

If you want to give certain users the permission to use the *Auditor Timeline*, create a new pseudo table in AppGini. Choose your own name for that table (if you copy the code below, name it *auditortimeline*). Once you generated your application, edit the generated file in the root (!) folder of your application.

Note: This file will be overwritten when you regenerate your app. We suggest saving the file to a different folder, so that you can simply copy it back later.

Note: You will need to give access permissions to the group for table *auditortimeline*! **Warning:** Every person (usergroup) having access to this timeline can view all records in all tables!

Replace *all* code in the file and copy and paste the following code into the file:

```
<?php
session_start();
$currDir = dirname(__FILE__);
include("$currDir/defaultLang.php");
include("$currDir/language.php");
include("$currDir/lib.php");

include_once("$currDir/header.php");

/* grant access to all users who have access to the orders table */
$user_can_access = get_sql_from('auditortimeline');
if(!$user_can_access) exit(error_message('Access denied!', false));

include("$currDir/hooks/audit/auditLog_timeline.php");
```

```
if(!$footerCode){
    include_once("$currDir/footer.php");
} else {
    ob_start(); include_once("$currDir/footer.php"); $dFooter=ob_get_contents(); ob
_end_clean();
    echo str_replace('<%%FOOTER%%>', $dFooter, $footerCode);
}
?>
```

Extras

Documenting changes for another table: fct 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.

Manual Installation (not recommended!)

Attention

Note 1

For the Search/Replace it's recommended to use 'Notepad++' available here: [Notepad++ Home Page](#)

Note 2

We 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, we suggest that you copy the hook files ONLY for the tables you wish to monitor into a separate directory and then BACKUP that directory. This way, you can do it speedily using Notepad++'s Search/Replace facility 'Find in Files' and do them all in just six shots.

Step 1. Extract the *auditlog_files.zip* and copy files

The zip file contain only 2 files:

- `auditLog_functions.php` : The functions that allow the audit log to work. Copy this into the `/hooks` folder of your application
- `auditLog.php` : A table (filterable and pageable) that will be added to the Admin Menu Options. Copy this into the `/admin` folder of your application.

Step 2. Create the Audit Log Table using the *audit_tableSQL.sql* file provided.

You may want to adjust the auditor tablename before. See [above](#): [Custom table name for the auditor table](#).

Then just run the SQL with your favorite tool (<https://www.phpmyadmin.net>, <https://www.adminer.org>).

Step 3. Essential File Modifications

3.1 Include audit-base files: '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 Add page to the Admin Menu Options: 'admin/incHeader.php'

Trick (as pictured and described in [Advanced Audit Log Table \(recommended\)](#), above):
 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 no one can change anything in that table.
 If you do 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><ul class="nav navbar-nav">  
  <li class="dropdown">  
    <a href="#" class="dropdown-toggle" data-  
toggle="dropdown"><i class="glyphicon glyphicon-tasks"></i> Audit Log  
    <b class="caret"></b></a>  
    <ul class="dropdown-menu">  
      <li><a href="auditLogTimeline.php"><i  
class="glyphicon menu-item-icon text-info glyphicon-calendar"></i>  
Auditor Timeline</a></li>  
      <li><a href="auditLog.php"><i class="glyphicon menu-  
item-icon text-info glyphicon-th"></i> Auditor Table</a></li>  
    </ul>  
  </li>  
</ul>
```

Step 4. Essential /hooks/folder-files modification

After you've read [Note 3](#), above!

In the *temp* folder that contains the files from the hooks-folder for *all the tables that you wish to monitor*, make the following changes to all these files. Recommended: Do 'find in files'. Code changes/additions are **color coded like this**.

A. Do 'find in files' for: (Remember to set the correct directory!)

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

and replace with:

```
init(&$options, $memberInfo, &$args){  
  $_SESSION ['tablename'] = $options->TableName;
```

```
$_SESSION ['tableID'] = $options->PrimaryKey;
```

B. Do 'find in files' for:

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

and replace with:

```
after_insert($data, $memberInfo, &$amp;args){  
table_after_change ($_SESSION, $memberInfo, $data, 'INSERTION');
```

C. Do 'find in files' for:

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

and replace with:

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

D. Do 'find in files' for:

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

and replace with:

```
after_update($data, $memberInfo, &$amp;args){  
table_after_change ($_SESSION, $memberInfo, $data, 'UPDATE');
```

E. Do 'find in files' for:

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

and replace with:

```
before_delete($selectedID, &$skipChecks, $memberInfo, &$amp;args){  
table_before_change($_SESSION, $selectedID);
```

F. Do 'find in files' for:

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

and replace with:

```
after_delete($selectedID, $memberInfo, &$amp;args){  
table_after_change ($_SESSION, $memberInfo, $selectedID, 'DELETION');
```

Remember to copy the files from the temp directory created in [Note 3](#) back to the original Hooks folder!

AppGini Code for Auditor table

File Tables Project Add-ons Help

75 tables in database.

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040

```

Created by Olaf Nöhring, 2021-03-02, <https://datenbank-projekt.de>

- changed SQL for auditor table. res_id should be (and is now) varchar to accommodate for non integer primary keys.
- Added res_id field fo internal auditLog table (which is not recommended to be used, see [Advanced Audit Log Table \(recommended\)](#))

Adjustments by Olaf Nöhring, 2021-02 (<https://datenbank-projekt.de>) for v1.77:

- corrected manual installation table_after_change call, using \$selectedID instead of \$data (which does not exist at this point)
- adjusted function table_after_change to accept \$selectedID as direct variable (not array)
- corrected error when inserting a new record and calling table_before_change to document all values

Adjustments by Olaf Nöhring, 2021-02 (<https://datenbank-projekt.de>) for v1.76:

- Fixed and updated manual installation _init code
- added [AppGini Code for Auditor table](#) for easy, quick and correct creation of the auditor table in your AppGini application.
- cooperation with landinialejandro to make code cleaner for use in installation plugin

Adjustments by Olaf Nöhring, 2021-01 (<https://datenbank-projekt.de>) for v1.74:

- Added example screenshots of Audit Log view to docs
- Added hint how to deal with the need for exact position of audit-log calls

Adjustments by Olaf Nöhring, 2021-01 (<https://datenbank-projekt.de>) for v1.73:

- Adjusted docs for more clarity

Adjustments by Olaf Nöhring, 2021-01 (<https://datenbank-projekt.de>) for v1.72:

- Added link to docs for the wonderful plugin extension from [landinialejandro](#) which makes installation a walk in the park. Please see below
- Modified file structure of the zip files that holds the audit log to adjust for use in combination with the plugin.
- Changed formatting of the docs for better readability
- Restructured docs for plugin
- Added hint for auditor tablename to docs

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 the 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!