

Intégration Base de données

? Guide d'Intégration Base de Données - Admin Menu v2.1

Auteur: ProtoMehka **Version:** 2.1 **Date:** 10 Janvier 2025 **Framework:** Altis Life 5.0+

? Introduction

Ce guide détaille la configuration de la **base de données** pour le menu admin v2.1. Il couvre :

- La table `admin_logs` pour l'historique des actions
- Les modifications de la structure existante
- Les requêtes SQL nécessaires

Prérequis :

- Base de données MySQL/MariaDB fonctionnelle
 - Accès phpMyAdmin ou ligne de commande MySQL
 - Table `pLayers` existante
-

?? Structure de la Base de Données

Table: `admin_logs`

Cette table enregistre toutes les actions effectuées par les admins.

Schéma Complet

```
CREATE TABLE IF NOT EXISTS `admin_logs` (  
  `id`          INT NOT NULL AUTO_INCREMENT,  
  `pid`        VARCHAR(17) NOT NULL,
```

```

`player_name`    VARCHAR(32) NOT NULL,
`target`        VARCHAR(64) DEFAULT NULL,
`execution_time`  TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
`command`       VARCHAR(64) NOT NULL,
`details`       TEXT,

PRIMARY KEY (`id`),
INDEX `index_pid` (`pid`),
INDEX `index_command` (`command`),
INDEX `index_execution_time` (`execution_time`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

Description des Colonnes

Colonne	Type	Nullable	Description
<code>id</code>	INT	Non	ID unique auto-incrémenté
<code>pid</code>	VARCHAR(17)	Non	Steam ID de l'admin (76561198...)
<code>player_name</code>	VARCHAR(32)	Non	Nom du joueur admin
<code>target</code>	VARCHAR(64)	Oui	Nom du joueur ciblé (si applicable)
<code>execution_time</code>	TIMESTAMP	Non	Date/heure de l'action (auto)
<code>command</code>	VARCHAR(64)	Non	Type d'action (TELEPORT, GODMODE, etc.)
<code>details</code>	TEXT	Oui	Informations additionnelles (coordonnées, montant, etc.)

Index

- **PRIMARY KEY** sur `id` : Identifiant unique
- **INDEX** sur `pid` : Recherche rapide par admin
- **INDEX** sur `command` : Filtrage par type d'action
- **INDEX** sur `execution_time` : Tri chronologique optimisé

? Installation

Méthode 1: Script SQL Complet

Fichier : altislife.sql (fourni)

Exécutez le fichier SQL complet qui contient la table admin_logs avec toutes les autres tables du serveur.

Via phpMyAdmin :

1. Sélectionner la base de données
2. Onglet "Importer"
3. Choisir altislife.sql
4. Cliquer "Exécuter"

Via ligne de commande :

```
mysql -u username -p database_name < altislife.sql
```

Méthode 2: Ajout Manuel de la Table

Si vous avez déjà une base de données existante :

```
CREATE TABLE IF NOT EXISTS `admin_logs` (  
  `id`          INT NOT NULL AUTO_INCREMENT,  
  `pid`         VARCHAR(17) NOT NULL,  
  `player_name` VARCHAR(32) NOT NULL,  
  `target`      VARCHAR(64) DEFAULT NULL,  
  `execution_time` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `command`     VARCHAR(64) NOT NULL,  
  `details`     TEXT,  
  
  PRIMARY KEY (`id`),  
  INDEX `index_pid` (`pid`),  
  INDEX `index_command` (`command`),  
  INDEX `index_execution_time` (`execution_time`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

? Types d'Actions Loggées

Liste des Commandes

Commande	Cible	Détails	Exemple
COMPENSATE	Nom joueur	Montant	Amount: \$50000
TELEPORT	-	Coordonnées	X:3612 Y:13138
TP_HERE	Nom joueur	-	-
GODMODE	-	État	ON ou OFF
FREEZE	Nom joueur	-	-
SPECTATE	Nom joueur	-	-
REPAIR_OBJECT	-	Type objet	Type: C_Offroad_01_F
DELETE_OBJECT	-	Type objet	Type: C_Offroad_01_F
DELETE_ITEM	-	Type et classe	Type: weapon Class: hgun_Rook40_F
SERVER_MESSAGE	-	Message	Texte du message
DEBUG_CONSOLE	-	-	-
MARKERS	-	État	ON ou OFF

Exemples de Données

```
-- Teleport
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'TELEPORT', NULL, 'X:3612 Y:13138');

-- Compensation
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'COMPENSATE', 'PlayerName', 'Amount: $50000');

-- GodMode ON
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'GODMODE', NULL, 'ON');

-- Freeze Player
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'FREEZE', 'TargetPlayer', NULL);

-- Delete Item (v2.1)
INSERT INTO admin_logs (pid, player_name, command, target, details)
VALUES ('76561198012345678', 'AdminName', 'DELETE_ITEM', NULL, 'Type: weapon | Class:
hgun_Rook40_F');
```

? Requêtes Utiles

Voir les Dernières Actions

```
SELECT
    id,
    player_name AS Admin,
    command AS Action,
    target AS Cible,
    details AS Détails,
    execution_time AS Date
FROM admin_logs
ORDER BY execution_time DESC
LIMIT 50;
```

Filtrer par Admin

```
SELECT * FROM admin_logs
WHERE pid = '76561198012345678'
ORDER BY execution_time DESC;
```

Filtrer par Type d'Action

```
SELECT * FROM admin_logs
WHERE command = 'TELEPORT'
ORDER BY execution_time DESC
LIMIT 20;
```

Actions sur un Joueur Spécifique

```
SELECT * FROM admin_logs
WHERE target = 'PlayerName'
ORDER BY execution_time DESC;
```

Statistiques par Admin

```
SELECT
    player_name AS Admin,
    COUNT(*) AS Total_Actions,
    COUNT(DISTINCT command) AS Types_Actions
FROM admin_logs
GROUP BY player_name
ORDER BY Total_Actions DESC;
```

Statistiques par Type d'Action

```
SELECT
    command AS Action,
    COUNT(*) AS Nombre
FROM admin_logs
GROUP BY command
ORDER BY Nombre DESC;
```

Actions des Dernières 24h

```
SELECT * FROM admin_logs
WHERE execution_time >= NOW() - INTERVAL 24 HOUR
ORDER BY execution_time DESC;
```

?? Maintenance

Nettoyage des Vieux Logs

Pour éviter que la table ne devienne trop volumineuse :

Supprimer les logs de plus de 30 jours :

```
DELETE FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 30 DAY;
```

Supprimer les logs de plus de 90 jours :

```
DELETE FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 90 DAY;
```

Archivage (Recommandé)

Au lieu de supprimer, créer une table d'archive :

```
-- Créer la table d'archive
CREATE TABLE IF NOT EXISTS `admin_logs_archive` LIKE `admin_logs`;

-- Copier les vieux logs
INSERT INTO admin_logs_archive
SELECT * FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 90 DAY;

-- Supprimer de la table principale
DELETE FROM admin_logs
WHERE execution_time < NOW() - INTERVAL 90 DAY;
```

Optimisation de la Table

Après de nombreux suppressions :

```
OPTIMIZE TABLE admin_logs;
```

? Surveillance

Taille de la Table

```
SELECT
    table_name AS `Table`,
    ROUND(((data_length + index_length) / 1024 / 1024), 2) AS `Size (MB)`
FROM information_schema.TABLES
WHERE table_schema = 'altislife'
```

```
AND table_name = 'admin_logs';
```

Nombre de Logs

```
SELECT COUNT(*) AS Total_Logs FROM admin_logs;
```

Logs par Jour (7 derniers jours)

```
SELECT
    DATE(execution_time) AS Jour,
    COUNT(*) AS Nombre_Actions
FROM admin_logs
WHERE execution_time >= NOW() - INTERVAL 7 DAY
GROUP BY DATE(execution_time)
ORDER BY Jour DESC;
```

? Sécurité

Permissions Recommandées

L'utilisateur MySQL du serveur Arma devrait avoir uniquement :

```
GRANT SELECT, INSERT ON altislife.admin_logs TO 'arma_user'@'localhost';
```

Ne pas donner :

- `DELETE` - Pour éviter suppression accidentelle
- `UPDATE` - Les logs ne doivent jamais être modifiés
- `DROP` - Protection contre suppression de table

Sauvegarde

Sauvegarder régulièrement la table :

```
mysqldump -u username -p altislife admin_logs > admin_logs_backup_$(date +%Y%m%d).sql
```

Restaurer depuis une sauvegarde :

```
mysql -u username -p altislife < admin_logs_backup_20251007.sql
```

? Tests

Test 1: Création de la Table

```
SHOW CREATE TABLE admin_logs;
```

Résultat attendu : Affiche la structure complète de la table

Test 2: Insertion Manuelle

```
INSERT INTO admin_logs (pid, player_name, command, target, details)  
VALUES ('76561198012345678', 'TestAdmin', 'TEST', 'TestTarget', 'Test details');
```

Vérification :

```
SELECT * FROM admin_logs WHERE command = 'TEST';
```

Test 3: Requête de Lecture

```
SELECT * FROM admin_logs ORDER BY execution_time DESC LIMIT 10;
```

Résultat attendu : Les 10 derniers logs

? Checklist d'Installation

- Table admin_logs créée
- Index créés (pid, command, execution_time)
- Charset utf8mb4 configuré
- Permissions utilisateur correctes
- Sauvegarde configurée
- Test d'insertion réussi
- Test de lecture réussi

? Nouveautés v2.1

Nouvelles Actions Loggées

Object Manager :

- `DELETE_ITEM` - Suppression d'items individuels depuis le gestionnaire d'objets
- Format des détails : `Type: weapon | Class: hgun_Rook40_F`
- Types possibles : `weapon`, `magazine`, `item`, `backpack`

Exemple de requête pour Object Manager :

```
SELECT * FROM admin_logs
WHERE command = 'DELETE_ITEM'
ORDER BY execution_time DESC
LIMIT 20;
```

Actions Admin avec Logging Renforcé :

- Toutes les commandes admin (11 au total + Object Manager)
- Logs automatiques avec timestamp
- Aucune modification de structure nécessaire

Compatibilité

La structure de la table `admin_logs` v2.0 est **100% compatible** avec v2.1 :

- Aucune modification de schéma requise
- Les nouvelles actions utilisent les colonnes existantes
- Migration transparente depuis v2.0

? Exemple de Rapport

Rapport Hebdomadaire

```
SELECT
    player_name AS Admin,
```

```
command AS Action,  
COUNT(*) AS Total,  
DATE(execution_time) AS Jour  
FROM admin_logs  
WHERE execution_time >= NOW() - INTERVAL 7 DAY  
GROUP BY player_name, command, DATE(execution_time)  
ORDER BY execution_time DESC;
```

Top 10 Admins Actifs

```
SELECT  
player_name AS Admin,  
COUNT(*) AS Total_Actions  
FROM admin_logs  
WHERE execution_time >= NOW() - INTERVAL 30 DAY  
GROUP BY player_name  
ORDER BY Total_Actions DESC  
LIMIT 10;
```

Installation base de données terminée ! ☑☑

Le système de logging est maintenant complètement opérationnel.

Version 2.1 - 10 Janvier 2025

Revision #3

Created 7 October 2025 13:25:25 by ProtoMehka

Updated 10 October 2025 14:26:27 by ProtoMehka