

# GLPI

- GLPI Plugin Datenbank initialisieren
- GLPI Plugin programmieren

# GLPI Plugin Datenbank initialisieren

## Einleitung

Wenn wir Daten in GLPI permanent machen wollen, müssen wir dies über Datenbanken realisieren. Da GLPI von Haus aus eine Datenbank benötigt, wollen wir diese verwenden, um dort unsere Datei abzulegen.

## hook.php

Um unsere Plugin-Datenbank zu initialisieren, müssen wir in der **install** Funktion ein paar Zeilen Code hinzufügen und diesen mit unserem SQL Befehl ausstatten.

So könnte der Code aussehen.

```
<?php
function plugin_debitors_install()
{
    global $DB;

    $migrate = New Migration(100);

    //Überprüfen ob Tabelle existiert
    if(!$DB->tableExists('glpi_plugin_debitors_debitors'))
    {
        $query = "CREATE TABLE glpi_plugin_debitors_debitors (id INT, name VARCHAR(50))";

        $DB->queryOrDie($query, $DB->error());
    }

    $migrate->executeMigration();
    return true;
}

function plugin_debitors_uninstall() {
    global $DB;
```

```
$tables = [  
    'debtors'  
];  
  
foreach ($tables as $table) {  
    $tablename = 'glpi_plugin_debtors_' . $table;  
  
    if ($DB->tableExists($tablename)) {  
        $DB->queryOrDie(  
            "DROP TABLE `{$tablename}`",  
            $DB->error()  
        );  
    }  
}  
  
return true;  
}
```

# GLPI Plugin programmieren

## Einleitung

In dieser und weiter Anleitung werden wir ein Plugin für **GLPI** schreiben. Das Plugin wird dann in der **Plugin Übersicht** angezeigt. Weitere Funktionen fügen wir in anderen Anleitungen hinzu.

Mithilfe von Plugins können wir unsere Anwendung anpassen und weitere Funktionalitäten hinzufügen, damit die Software unseren Anforderungen gerecht wird.

## setup.php & hook.php

In dieser Anleitung werden wir die **setup.php** und die **hook.php** einrichten. Diese beiden Dateien sind eine der wichtigsten. Anhand dieser Dateien werden die Plugins identifiziert und installiert als auch später wieder deinstalliert.

Zuerst wechseln wir in das **plugin** Verzeichnis unseres **GLPI-Servers**. Dort erstellen wir einen Ordner mit dem Namen unseres Plugins. Wir nennen unser Plugin **debtors**.

Mit dem Plugin soll es möglich sein, Debitoren hinzuzufügen und diese später mit bestimmten Daten zu verknüpfen.

In dem Verzeichnis erstellen wir eine Datei mit dem Namen **setup.php**. Dort fügen wir folgendes Code Gerüst ein.

```
<?php
// Definiere Variablen mit Plugin Informationen
define("PLUGIN_NAME", "Debitoren");
define("VERSION_NUMBER", "1.0.0");

// Erstelle Initialisierungsfunktion des Plugins
function plugin_init_Debitors()
{
    global $PLUGIN_HOOKS;
    $PLUGIN_HOOKS['csrf_compliant']['Debtors'] = true;
}

// Definiere Plugin Informationen
function plugin_version_debtors()
```

```

{
    return array(
        "name" => PLUGIN_NAME,
        "version" => VERSION_NUMBER,
        "author"=> "DEIN NAME",
        "license" => "LIZENZ",
        "homepage" => "DEINE WEBSITE",
        "minGlpiVersion" => "VERSIONS ANFORDERUNG"
    );
}

//Überprüfe ob Anforderungen für das Plugin erfüllt sind
function plugin_debitors_check_prerequisites()
{
    if(version_compare(GLPI_VERSION, '10', 'eq') || version_compare(GLPI_VERSION, '10', 'gt'))
    {
        return true;
    } else {
        echo "Das Plugin benötigt die Version 10.0.0 | Installiert ist Version: " . GLPI_VERSION;
        return false;
    }
}

//Überprüfe ob die Konfiguration übereinstimmt
function plugin_debitors_check_config()
{
    return true;
}

?>

```

In der Datei brauchen wir nur noch den Namen, und die anderen Informationen ändern. Diese Informationen werden dann im **Plugin Menü** angezeigt. Im Weiteren können wir in der **check\_prerequisites** Funktion die Anforderungen an das System Überprüfen, ob das Plugin überhaupt installiert werden darf.

Als letzten Schritt für dieses Kapitel erstellen wir eine Datei mit dem Namen **hook.php**. In diese Datei fügen wir folgenden Inhalt ein:

```

<?php
function plugin_debitors_install()
{

```

```
    return true;  
}
```

```
function plugin_debitors_uninstall()  
{  
    return true;  
}
```

In diesen Funktionen können wir sämtlichen Code hineinschreiben, der ausgeführt wird, wenn wir das Plugin installieren oder deinstallieren.

Wenn wir jetzt das Plugin Menü aufrufen, können wir unser Plugin sehen und installieren.