

# **Tutorial zur Verbindung von BlitzBasic und PHP**

von Ofenrohr

## Worum geht es?

In diesem Tutorial möchte ich einen einfachen und leicht erweiterbaren Weg aufzeigen um PHP und BlitzBasic miteinander zu verbinden.

Der Vorteil der Verbindung besteht darin, dass man mithilfe von PHP online Highscores, Benutzerverwaltungen oder weitaus komplexere Dinge erstellen kann. PHP stellt in diesem Fall lediglich den Server dar. BlitzBasic hingegen ist für die Eingaben bzw. für das Spiel selbst verantwortlich.

Um den Spielspass eines Spiels zu erhöhen besteht z.B. die Möglichkeit online Highscores in das Spiel zu integrieren.

In diesem Tutorial werde ich einige simple Möglichkeiten zeigen, wie man einen online Highscore verwirklicht.

## Was brauche ich?

Um einen online Highscore zu haben braucht man erstmal ein Spiel, für das die Highscore sein soll.

Man sollte außerdem im Besitz eines Webspace sein, der MySQL Datenbanken unterstützt.

Elementare Kenntnisse von PHP und MySQL sind empfehlenswert. In diesem Tutorial werde ich jeden Code erklären, sodass diese Kenntnisse nicht unbedingt notwendig sind. Wer aber eine Schummelsichere Highscore erstellen will, sollte sich mit PHP ein wenig auskennen um z.b. Verschlüsselung oder ähnliches zu integrieren.

Elementare Grundkenntnisse erhält man auf folgender Seite:

<http://www.schattenbaum.net/php/>

Befehlsreferenzen sind hier zu finden:

<http://www.php.net/manual/de/>

<http://de.selfhtml.org/>

## Wie funktioniert die Datenübertragung?

Es gibt zwei Instanzen. Das PHP Script soll Daten in eine MySQL Datenbank schreiben, die es vom BlitzBasic Programm erhält.

## Aufbau in PHP

In PHP gibt es 2 simple Möglichkeiten um Daten von außen zu empfangen. Sie heißen GET und POST. Wer sich ein wenig mit html auskennt, wird wissen, das man bei Formularen bei der Method zwischen GET und POST wählen kann. In diesem Tutorial werde ich die GET

Methode benutzen und erklären. Wenn man aber sicher sein möchte, dass es nicht so leicht ist zu schummeln, sollte man auf jeden Fall die POST Methode benutzen.

#### Aufbau in Blitzbasic

In BlitzBasic muss man einfach nur die Daten an das PHP Script übertragen und schauen, was das Script zurück liefert. Um ein PHP Script aufzurufen muss man in BlitzBasic beim Aufruf einen Bestimmten Text absenden, damit man die gewünschten Daten vom Server gesendet bekommt. Dieser spezielle Text ist nach dem HTTP 1.1 Protokoll aufgebaut. HTTP steht für Hyper Text Transfer Protokoll ([RFC 2616](#)) Wer sich erinnert: Beim Browser schreibt man immer http:// vor die Adresse. Dies verrät dem Browser welches Protokoll er benutzen soll. Es gibt einige Downloads, die über FTP (File Transfer Protokoll) ablaufen. Diese haben ein ftp:// vor der Adresse.

#### Vorbereitungen in PHP

Um die Daten des PHP Script zu speichern, benutzt man am besten eine MySQL Datenbank. Bei fast jedem Webspaces ist ein MySQL Datenbank Admin Tool dabei. Es ist eigentlich immer PHPmyAdmin. Dieses Web Tool ist leicht zu benutzen.

Man logt sich in PHPmyAdmin ein und erstellt in einer Datenbank eine Tabelle.



Wir nennen diese Tabelle: highscore. MySQL Datenbanken sind vergleichbar mit Types in Blitzbasic. Sie enthalten einen Grundaufbau von Variablen. Genau wie Types können sie beliebig viele Einträge enthalten. Die Verwaltung von MySQL Datenbanken ist aber wesentlich bequemer.

Unsere Tabelle highscore soll den Usernamen und seine Punktzahl enthalten. Außerdem brauchen wir noch eine ID an der wir jeden Eintrag einwandfrei identifizieren können. Also wird unsere Tabelle 3 Felder enthalten.

Wenn wir die Werte eingetragen haben (siehe Bild), dann kommen wir zu einer Oberfläche, in der wir wesentlich mehr Angaben machen müssen.

Hierzu habe ich ebenfalls einen Screenshot gemacht.

Server: localhost ▶ Datenbank: usr\_web4\_1 ▶ Tabelle: highscore

Feld	Typ	Länge/Set	Attribute	Null	Standard**	Extra					
	VARCHAR			not null							
	VARCHAR			not null							
	VARCHAR			not null							

Tabellen-Kommentar:  Tabellentyp: Standard

1

\* Wenn das Feld vom Typ 'ENUM' oder 'SET' ist, benutzen Sie bitte das Format: 'a','b','c',....  
 Wann immer Sie ein Backslash ("\") oder ein einfaches Anführungszeichen (") verwenden,  
 setzen Sie bitte ein Backslash vor das Zeichen. (z.B.: '\xyz' or 'a\b').

\*\* Bitte geben Sie jeweils nur einen Standardwert ohne Escape- oder Anführungszeichen an.

Ganz links sieht man den Eintrag „Feld“.

Hier werden die Namen der Variablen eingetragen. Die erste Variable heißt „id“. Sie dient, wie schon oben gesagt, zur genauen Identifizierung eines Eintrags.

Die zweite Variable heißt „name“. Damit ist der Name des Spielers gemeint.

Im letzte Variable nennen wir „punkte“. In ihr wird der Punktestand des Spielers gespeichert.

Gleich neben dem Eintrag „Feld“ sehen wir „Typ“. Hier wird der Variablen Typ des Eintrags bestimmt.

Für die „id“ Variable nehmen wir den Typ „Int“ und im daneben liegenden Feld für die Länge „6“.

„Name“ hat den Typ Text und „punkte“ wird „BIGINT“ mit der Länge „12“.

Außerdem wird dem Feld „id“ der „Primary-Key“ zugewiesen.



Dadurch ist gesichert, das keine zwei Einträge den gleichen ID Wert haben können. Um das Eintragen der ID zu erleichtern, wählen wir unter Extra für ID „auto\_increment“ aus. Dadurch wird immer die richtige ID zugewiesen.

Zum Abschluss drücken wir auf „Speichern“ und sehen uns das Ergebnis an.

	Feld	Typ	Attribute	Null	Standard	Extra	Aktion
<input type="checkbox"/>	id	int(6)		Nein		auto_increment	
<input type="checkbox"/>	name	text		Nein			
<input type="checkbox"/>	punkte	bigint(12)		Nein	0		

Dies ist ein Teil des Bildes, das wir nach dem drücken auf den Speichern Button erhalten.

Damit wäre der Teil in der MySQL Datenbank erledigt.

Wenden wir uns nun dem eigentlichen PHP Script zu. Mit diesem Script wollen wir auf die eben erstellte MySQL Datenbank zugreifen und neue Einträge erstellen.

Dafür erstellen wir erstmal ein neues PHP Script auf dem Server.

```
<?PHP
    @mysql_connect("localhost", "---User---","---Passwort---") or die ("0");
    @mysql_select_db("usr_web4_1") or die ("1");
    // usr_web4_1 <- muss durch eure Datenbank ersetzt werden
?>
```

Erklärung:

@mysql\_connect("localhost", "---User---","---Passwort---") verbindet zum Server. Das @ vor mysql\_connect sorgt dafür, dass keine Fehlermeldung ausgegeben wird, falls man falsche angaben macht. Stattdessen sorgt or die ("0"); dafür, dass, falls ein Fehler auftritt einfach eine 0 zurückgeliefert wird. Dies ist in Blitzbasic einfacher zu Verarbeiten. "---User---" und "---Passwort---" ersetzt ihr mit den Werten, mit denen ihr euch auch in PHPmyAdmin eingeloggt habt.

@mysql\_select\_db("usr\_web4\_1") or die ("1"); ist genau so aufgebaut wie der vorherige Befehl, bloß dass hier die Datenbank ausgewählt wird. In meinem Fall heißt die Datenbank usr\_web4\_1. Mit sehr großer Wahrscheinlichkeit heißt sie bei euch anders. Deshalb müsst ihr bei euch den passenden Wert eintragen.

Nun aber mal zur Erklärung von GET. Mit GET schreibt man einfach die Variablen, die übergeben werden sollen hinten an das Script ran. Das sieht dann wie folgt aus:

[http://www.server.de/script.php?variable=text&variable2=hallo\\_welt](http://www.server.de/script.php?variable=text&variable2=hallo_welt)

Wie man am Beispiel sehen kann ist, dass alle Variablen mit einem ? angehängt werden. Zuerst kommt die Variablenbezeichnung ein = Zeichen und anschließend der Wert. Dies kann man beliebig oft wiederholen, dann allerdings nicht mehr mit einem ? sondern mit einem &. Es ist ersichtlich einfach auf diese Weise Daten ans Script zu übertragen. Doch dazu später noch genaueres.

Jetzt werde ich erklären, wie man in PHP den Wert der Übergebenen Variablen abrufen kann.

Hierzu auch wieder ein simples Code Beispiel:

```
<?PHP
    $name=$_GET['name'];
    $punkte=$_GET['punkte'];
    echo($name.“ hat „.$punkte.“ erreicht“);
?>
```

Wir gehen jetzt einfach davon aus, dass das Script highscore.php heißt und wie folgt aufgerufen wurde:

highscore.php?name=ofenrohr&punkte=12345

Wie man unschwer erkennen kann, liefert \$\_GET['variablen\_name']; den in der URL angegebenen Wert zurück.

In unserem Beispiel wurde name und punkte übergeben. Diese Werte wurden mithilfe von \$\_GET an \$name und \$punkte übergeben.

Somit ist auch erklärt, wie die Daten an das PHP Script werden. Nun zur Speicherung selbiger in die MySQL Datenbank.

Hierfür nehmen wir wieder unser altes Beispiel mit der Verbindung zur MySQL Datenbank und verbinden es mit dem neuen. Anschließend werden noch ein wenig MySQL hinzugefügt:

```
<?PHP
    @mysql_connect("localhost", "---User---", "---Passwort---") or die ("0");
    @mysql_select_db("usr_web4_1") or die ("1");
    // usr_web4_1 <- muss durch eure Datenbank ersetzt werden

    // abfrage der Daten
    $name=$_GET['name'];
    $punkte=$_GET['punkte'];

    // einen neuen Eintrag in die MySQL Datenbank machen
    $sql="INSERT INTO highscore (name,punkte) VALUES ('$name','$punkte)";
    mysql_query($sql);

    // eine Bestätigung ausgeben
    die("2");
?>
```

Erklärung:

```
$sql="INSERT INTO highscore (id,name,punkte) VALUES ('$id','$name','$punkte')";
```

Dieser MySQL Befehl erklärt sich fast von selbst. INSERT INTO bedeutet soviel wie: „Füge ein in“ highscore ist wieder unsere Tabelle. (id,name,punkte) sind die Felder, die Werte erhalten werden. VALUES ('\$id','\$name','\$punkte') gibt die Werte in der Reihenfolge an, wie sie vorher deklariert wurde. \$id ist unsere neue größte ID. \$name und \$punkte sind die Informationen, die mit der GET Methode abgefragt haben.

```
mysql_query($sql);
```

Dieser Befehl sendet einfach nur den MySQL Befehl ab.

Und zum Schluss noch eine Bestätigung für das Spiel, das auch alles gut gelaufen ist: die("2");

Der „die“ (zu Deutsch: Stirb) Befehl beendet das PHP Script vollständig und gibt vorher noch den angegebenen Text aus.

Hier noch zwei kleine Bilder, wie es aussehen könnte:

Die Testeingabe im Browser:



Wenn alles richtig von statten gegangen ist, wird eine 2 im Browser Fenster erscheinen.

In der MySQL Datenbank können sie sich den Inhalt einer Tabelle anzeigen lassen, indem sie die Tabelle auswählen und dann auf „Anzeigen“ klicken.



	id	name	punkte
	1	ofenrohr	12345

Wenn das alles funktioniert kann man bereits Einträge hinzufügen, aber man hat noch keine Möglichkeit sie auszugeben.

Ich werde mich nicht mit Design aufhalten, weil es nichts zur Sache tut. Ich werde hier eine simple Möglichkeit zum Ausgeben der Highscore zeigen.

Dafür erstellen wir ein neues PHP Script. Der Inhalt könnte wie folgt aussehen:

```

<html>
<head>
<title>Highscore von SPIEL</title>
</head>
<body>
<table border="2" align="center" width="500">
<tr>
<td colspan="2"><div align="center">Highscore</div></td>
</tr>
<tr>
<td width="60%"><div align="center">Username</div></td>
<td width="40%"><div align="center">Punkte</div></td>
</tr>
<?
    // genau wie in highscore.php muss man sich zur Datenbank verbinden
    @mysql_connect("localhost", "---User---", "---Passwort---") or die ("Fehler beim Verbinden
zur Datenbank!");
    @mysql_select_db("usr_web4_1") or die ("Fehler beim Verbinden zur Datenbank!");

    // Daten abrufen
    $sql="SELECT name,punkte FROM highscore ORDER BY punkte DESC";
    $ergebnis=mysql_query($sql);

    // Daten ausgeben
    while ($row=mysql_fetch_object($ergebnis)) {
        echo('<tr>');
        echo('<td width="60%"><div align="center">'.$row->name.'</div></td>');
        echo('<td width="40%"><div align="center">'.$row->punkte.'</div></td>');
        echo('</tr>');
    }
?>
</table>
</body>
</html>

```

#### Erklärung:

Ich werde nicht den HTML Teil erklären, weil ich einfach erwarte, das ihr das könnt, weil HTML sehr leicht zu erlernen ist. Falls trotzdem mal Fragen zu einigen HTML bestehen sollten empfehle ich folgende Seite: <http://de.selfhtml.org/>

Genau wie in highscore.php müssen wir zuerst eine Verbindung zur Datenbank herstellen. Der einzige Unterschied ist, das die Fehlermeldung jetzt in Normalem Text ausgegeben wird, den jeder Verstehen kann.

```
$sql="SELECT name,punkte FROM highscore ORDER BY punkte DESC";
```

Dieser MySQL Befehl ruft den benötigten Inhalt aus highscores auf. Dies geschieht durch name,punkte. Um die Ausgabe ein wenig zu verschönern, wird der Spieler mit der Höchsten Punktzahl zuerst ausgegeben. Man kann auch Alphabetisch sortieren lassen, indem man ORDER BY name schreibt.

```
$ergebnis=mysql_query($sql);
```

Ist genau das gleiche wie in highscore.php und speichert das Ergebnis der Abfrage in \$ergebnis.

```
while ($row=mysql_fetch_object($ergebnis)) {
```

While ist englisch und bedeutet während. Solange mysql\_fetch\_object(\$ergebnis) zulässige Werte an \$row liefert, wird der Inhalt der schleife ausgeführt.

Der Inhalt der Schleife ist einfach die Ausgabe der Daten, die in der Datenbank liegen. Bei diesem Punkt möchte ich noch etwas zu den Strings und echo(); in PHP sagen. Es gibt zwei Möglichkeiten Strings mit echo auszugeben. Entweder, man schreibt: echo(„Hallo Welt“); oder man schreibt echo('Hallo Welt'); Auf den ersten Blick ist es egal, ob man einfache oder doppelte Anführungsstriche nimmt. Aber, wenn man in HTML z.B. <div align=“center“> benutzen will und das mit echo ausgeben muss, kann man einfache Anführungsstriche benutzen oder umständlich \“center\“ schreiben, was auf die Dauer recht nervig ist. Deshalb empfehle ich die Benutzung von einfachen Anführungszeichen.

Damit wäre der PHP Teil komplett. Probiert ein wenig damit herum, bis ihr den Code verstanden habt.

### Vorbereitungen in BlitzBasic

Da nun der PHP Server steht müssen wir uns nun nurnoch um den relativ kurzen Teil in Blitzbasic kümmern. Auf fast allen Servern ist das HTTP 1.1 Protokoll standard.

Ich werde hier ein simples Beispiel zeigen, wie man die Daten an das PHP Script übergibt.



```
name$=Input("Bitte Namen eingeben: ")
punkte%=int(input("Bitte geben sie ihre Punkte ein: "))
stream=open tcpstream("ofenrohr.chat-blitz.de",80)
if stream=0 then print "konnte nicht verbinden" : waitkey : end
writeline stream, "GET /test/highscore.php?name="+name+"&punkte="+punkte+" HTTP/1.1"
writeline stream, "HOST: ofenrohr.chat-blitz.de"
writeline stream, ""
while not eof(stream)
debuglog(readline(stream))
wend
close tcpstream(stream)
print "Score erfolgreich eingetragen..."
waitkey
end
```

#### Erklärung:

Ich gehe davor aus, das ihr BlitzBasic könnt, weshalb ich nur den HTTP Teil erklären werde.

Zuerst verbindet ihr euch auf den Server. http:// wird weggelassen, weil die eigentliche Adresse reicht. Der HTTP Transfer wird standartmäßig über Port 80 abgewickelt.

Wenn also die Verbindung hergestellt wurde, muss man angeben, welches Script ich aufrufen möchte. In diesem Fall also das Script highscore.php im Ordner /test/. Anschließend werden noch die Spielinformationen angegeben. Zum Abschluss muss man noch das Protokoll angeben. In unserem Fall also HTTP/1.1. Bei HOST: muss man nochmal die Adresse angeben, die man bereits beim Verbinden angegeben hat.

Es gibt noch einen großen Haufen weiterer Möglichkeiten, die man noch hinzufügen könnte. Man findet sie in dem tollen [RFC 2616](#). Wer keine Lust hat, Englisch zu lesen, hat Pech gehabt. Das meiste gute Zeug ist auf Englisch, da kommt man nicht herum.

Im Beispiel wird das was das PHP Script zurückliefert in die Debug Konsole ausgegeben. Falls irgentwas nicht klappt, wird dort auch die Fehlermeldung zu finden sein. Neben dem was das PHP Script ausgibt, wird auch ein Header da stehen. Dieser kann normalerweise einfach mit readline überlesen werden, weil er die selbe länge hat, es sei denn, im PHP Script wurden Cookies definiert, Sessions gestartet o.ä.

#### Abschluss

Ich möchte hier darauf hinweisen, das ich für eventuell entstandene Schäden, die durch dieses Dokument entstehen könnten nicht verantwortlich bin. Alles was sie tun, tun sie auf eigene Gefahr!

Dieses Dokument darf frei Kopiert werden, solange es nicht verändert wird und mein

Name auf der Kopie steht.

Fehler, Verbesserungsvorschläge o.ä. bitte an:

ICQ: 305779988

E-Mail: [petx\\_productions@yahoo.de](mailto:petx_productions@yahoo.de)