

# Blogeinträge

Für den Zugriff auf die Tabelle `blogentries`, also für das Erstellen und Anzeigen der Blogbeiträge erstellen wir jetzt eine Klasse `blogentry` in der Datei `class/blogentry.class.php`<sup>1)</sup>.



## (A1)

Das Basisgerüst der `blogentry`-Klasse hat dieselbe Funktionalität wie die `bloguser`-Klasse: Im Konstruktor muss das Handle für den Datenbankzugriff „geholt“ werden und als Objektvariable gespeichert werden.

Erstelle eine Datei mit diesem Basisgerüst, so dass in der `syntax.php` mit folgendem Code ein `blogentry`-Objekt instanziiert werden kann:

```
// Zugriff auf die Blogbeiträge
$blogentry = new blogentry($dbusername, $dbpasswd, $dbname, $dbhost);
```

Teste, ob dein Code fehlerfrei funktioniert.

## Hilfe

Die grundlegende `blogentry`-Klasse ist eine Kopie des Konstruktors der `bloguser`-Klasse:

```
<?php

class blogentry {
    protected $db;           // Das DB-Handle

    //Konstruktor
    public function __construct ($dbusername, $dbpassword, $dbname,
    $host="localhost" )
    {
        // PDO Connection erzeugen/holen und als
        // Objektattribut "speichern". Damit werden DB Zugriffe möglich.
        $this->db = mysqlldb::getConnection($dbusername, $dbpassword, $dbname,
    $host);
    }
}
```

?>

## Methoden der Blogentry-Klasse

Jetzt müssen wir die `blogentry`-Klasse um weitere Funktionalitäten erweitern. Dazu können wir uns zunächst überlegen, welche Funktionen die Klasse benötigt, und diese als „Interface“ definieren. Man kann sich vorstellen, dass ein Interface eine Klassendatei ist, die nur die Methodenköpfe ohne Code als „abstrakte Methoden“ enthält:

```
<?php
interface blogentryInterface
{
    // Alle Blogintraege anzeigen
    public function show_entries($userid=-1);

    // Neuen Blogeintrag hinzufügen
    public function insert_entry($userid);

    // Fehlaufrufe abfangen
    public function __call ($name, $param);
}
?>
```

Die Definition der `blogentry`-Klasse muss nun entsprechend angepasst werden:

```
class blogentry implements blogentryInterface {
    ...
}
```

[← Schritt 2](#) [Schritt 4 →](#)

1)

damit der Autoloader aus Schritt 1 sie findet...

From:  
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:  
[https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:projekt:dokuwiki\\_plugin:microblogging:step03:start?rev=1624295015](https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:projekt:dokuwiki_plugin:microblogging:step03:start?rev=1624295015)

Last update: 21.06.2021 19:03

