

# Neue Klassen mit Datenbankzugriff

Wir wollen unserem Projekt nun weitere Klassen hinzufügen, von denen einige Datenbankzugriff benötigen. Die jeweiligen Operationen auf der Datenbank sollen aber in den jeweiligen Klassen gekapselt sein.

Um dieses Ziel zu erreichen, *reduzieren* wir zunächst die Funktionalität unserer mysqlDb-Klasse darauf, eine <sup>1)</sup> Datenbankverbindung herzustellen und ein entsprechendes PDO-Objekt, das diese Verbindung repräsentiert, zurückzuliefern, welches dann innerhalb der anderen Klassen für den Datenbankzugriff verwendet werden kann.

## Die neue DB Klasse

Passe die Datenbank Klasse in der Datei mysqlDb.class.php folgendermaßen an:

```
class mysqlDb {

    static public $connection = null;
    private $dbobject;

    /**
     * Constructor - connect to db, set handle attribute
     *
     * @param string      $dbusername    DB username
     * @param string      $dbpassword    DB password
     * @param string      $dbname        Database to connect to
     * @param string      $host          Database host to connect to
     * (optional)
     *
     */

    private function __construct($dbusername, $dbpassword, $dbname, $host )
    {

        try {
            self::$connection = new PDO("mysql:host=$host;dbname=$dbname",
            "$dbusername", "$dbpassword");
            self::$connection->setAttribute(PDO::ATTR_ERRMODE,
            PDO::ERRMODE_EXCEPTION);

            } catch ( PDOException $e ) {
                echo 'Verbindung zur Datenbank fehlgeschlagen: ' .
                $e->getMessage();
                exit();
            }

        }

    }
```

```
public static function getConnection($dbusername, $dbpassword, $dbname,
$host ) {
    // Wenn das Objekt noch keine PDO-Instanz hat
    // wird eine erzeugt
    if(self::$connection === null) {
        $dbobject = new mysqli($dbusername, $dbpassword, $dbname,
$host);
    }

    //Und das PDO-Objekt zurückgeben
    return self::$connection;
}

}
```

#### Anmerkungen:

- Die Schreibweise mit dem Doppel-Doppelpunkt ermöglicht den Zugriff auf statische Methoden und Felder, ohne ein Objekt der Klasse zu besitzen. Das benutzen wir, um in weiteren Klassen ein Datenbank-Handle zu erhalten.

← Schritt 1Schritt 3 →

1)

und nur eine

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

Permanent link:  
[https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:projekt:dokuwiki\\_plugin:microblogging:step02:start?rev=1624287545](https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:datenbanken:projekt:dokuwiki_plugin:microblogging:step02:start?rev=1624287545)

Last update: 21.06.2021 16:59

