

```
<?php

/**
 * @author dominique.aigroz@edu.ge.ch
 * Modify by: Hoarau Nicolas
 */
require_once dirname(dirname(__DIR__)) . '/config/config.php';
require_once __DIR__ . '/ExtendedPdo.php';

/**
 * @brief Helper class encapsulating
 *        the PDO object
 * @author dominique.aigroz@kadeo.net
 * @remark
 */
class DatabaseController
{
    private static $pdoInstance;

    /**
     * @brief Class Constructor - Create a new database connection if one doesn't exist
     *        Set to private so no-one can create a new instance via ' = new
     *        KDatabase();'
     */
    private function __construct()
    {
    }

    /**
     * @brief Like the constructor, we make __clone private so nobody can clone the
     *        instance
     */
    private function __clone()
    {
    }

    /**
     * @brief Returns DB instance or create initial connection
     * @return $objInstance;
     */
    public static function getInstance()
    {
        if (!self::$pdoInstance) {
            try {
                $dsn = DB_DBTYPE . ':host=' . DB_HOST . ';port=' . DB_PORT . ';dbname='
                . DB_DBNAME . ';charset=utf8';
                self::$pdoInstance = new ExtendedPdo($dsn, DB_USER, DB_PASS);
                self::$pdoInstance->setAttribute(ExtendedPdo::ATTR_ERRMODE,
                ExtendedPdo::ERRMODE_EXCEPTION);
            } catch (PDOException $e) {
                echo "KDatabase Error: " . $e->getMessage();
            }
        }
        return self::$pdoInstance;
    }
}
```

```
51     }
52
53     # end method
54     /**
55      * @brief Passes on any static calls to this class onto the singleton PDO instance
56      * @param $chrMethod The method to call
57      * @param $arrArguments The method's parameters
58      * @return $mix The method's return value
59      */
60     final public static function __callStatic($chrMethod, $arrArguments)
61     {
62         $pdo = self::getInstance();
63         return call_user_func_array(array($pdo, $chrMethod), $arrArguments);
64     }
65
66     # end method
67 }
68
```