

Плейсхолдеры

- ?s («string») — строки (а также DATE, FLOAT и DECIMAL).
- ?i («integer») — целые числа.
- ?n («name») — имена полей и таблиц
- ?p («parsed») — для вставки уже обработанных частей запроса
- ?a («array») — набор значений для IN (строка вида 'a','b','c')
- ?u («update») — набор значений для SET (строка вида `field`='value',`field`='value')

Хелперы

- **parse()** — возвращает строку с обработанными плейсхолдерами. Для отладки и сборки сложных запросов
- **query()** — выполняет запрос и возвращает mysqli result. Может использоваться традиционно, с fetch() и т.д.
- **getOne()** — возвращает скаляр, первый элемент первой строки результата
- **getRow()** — одномерный массив, первую строку результата
- **getCol()** — одномерный массив скаляров — колонку таблицы
- **getAll()** — двумерный массив, индексированный числами по порядку
- **getInd()** — двумерный массив, индексированный значениями поля, указанного первым параметром
- **getIndCol()** — массив скаляров, индексированный полем из первого параметра. Для составления словарей вида key => value

Все хелперы вызываются одинаково: один обязательный параметр — запрос с плейсхолдерами, и сколько угодно опциональных параметров, количество и порядок которых должны совпадать с количеством и порядком расположения плейсхолдеров в запросе. У функций семейства Ind используется ещё один обязательный параметр — имя поля, по которому осуществляется индексация возвращаемого массива.

Стандартные методы, аналоги функций API

- **fetch(\$result, \$mode=self::RESULT_ASSOC)** - получает строку результата
- **affectedRows()** — возвращает количество строк, обработанных запросами DELETE или UPDATE
- **insertId()** — last insert id
- **numRows(\$result)** — количество строк в возвращённом результате. Нужно только для query(). Для остальных - count()
- **free(\$result)** — освобождает результат

Прочие функции

- **whiteList(\$input,\$allowed,\$default=FALSE)** — проверяет переданное значение по белому списку
- **filterArray(\$input,\$allowed)** — возвращает \$input, убрав из него все поля, отсутствующие в \$allowed
- **lastQuery()** — возвращает последний выполненный запрос
- **getStats()** — возвращает массив со всеми выполненными запросами и временем их выполнения

Примеры

```
// создание класса, коннект в БД
$db = new SafeMySQL(); // с настройками по умолчанию (root, пустой пароль, база test, utf8)
$db = new SafeMySQL(array('user' => 'user', 'pass' => 'pass', 'db' => 'db', 'charset' => 'cp1251'));

$data = $db->getInd('id','SELECT * FROM table WHERE cat=?s', $cat);

// сложная вставка. Основной массив данных лежит в $data, но IP адрес передается отдельно
$data = array('field'=>$value,'field2'=>$value);
$sql = 'INSERT INTO table SET ts=unix_timestamp(), ip=inet_aton(?s),?u';
$core->db->query($sql, $ip, $data);

// получение списка id и подстановка их в другую таблицу
$sids = $core->db->getCol('SELECT id FROM category WHERE root=?i',$root);
$cat = $core->db->getAll('SELECT * FROM cat WHERE category IN (?a)',$sids);

// INSERT .. ON DUPLICATE UPDATE
$data = array('offers_in' => $in, 'offers_out' => $out);
$sql = "INSERT INTO stats SET pid=?i,dt=CURDATE(),?u ON DUPLICATE KEY UPDATE ?u";
$db->query($sql,$pid,$data,$data);

// добавление части запроса по условию
$sqlpart = '';
if (!empty($var)) {
    $sqlpart = $db->parse(" AND field = ?s", $var);
}
$data = $db->getAll("SELECT * FROM table WHERE a=?i ?p", $id, $sqlpart);

// добавление нескольких условий
$w = array();
if ($one) $w[] = $db->parse("one = ?s", $one);
if ($two) $w[] = $db->parse("two IN (?a)", $two);
if ($tre) $w[] = $db->parse("tre <= ?i", $tre);
$where = "AND ".implode(' AND ', $w);
$data = $db->getAll("SELECT * FROM table WHERE 1 ?p LIMIT ?i,?i",$where, $start,$per_page);
```